

SONDERHEFT 73
Markt&Technik
öS 120,-/sfr. 16,-/Lit. 16000
hfl. 21,-/dkr. 75,-/fmk 62,-
DM 16,-
DISKETTE IM HEFT
64'er
Spiele
Alle Spiele auf Diskette
Action, Köpfchen, Strategie
10 Mega-Games
Programmierkurs
Von der Idee bis zum Spielstart
Gefahr aus dem All
Mission II: Endspurt gegen die Uhr
TILT
Brandheiße Schnupperversion mit drei Levels

AMIGA

die Nr.1

Das meistgekaufte Amiga-Magazin

Jetzt brandaktuell - die AMIGA-Ausgabe Nr.1 mit der perfekten Themenmischung:

Trends und Sensationen: Das werden die Hits '92!
Schauen Sie nach vorne: Mit AMIGA-Magazin erfahren Sie jetzt schon alles über neue Drucker, Erweiterungskarten, Software, Betriebssysteme und vieles mehr. 1992 wird es in sich haben...

Programmieren mit dem neuen 2.0!
Was kann das neue Betriebssystem 2.0 für den Amiga leisten? Was sind die Neuigkeiten, und wie werden sie programmiert? AMIGA-Magazin hat die Antworten.

Kaufberatung: Wie Sie Ihren Amiga sinnvoll erweitern!
Viele Hardware-Erweiterungen sorgen tatsächlich für mehr Effizienz. AMIGA-Magazin weiß, welche Produkte sich bewährt haben und welche nicht.

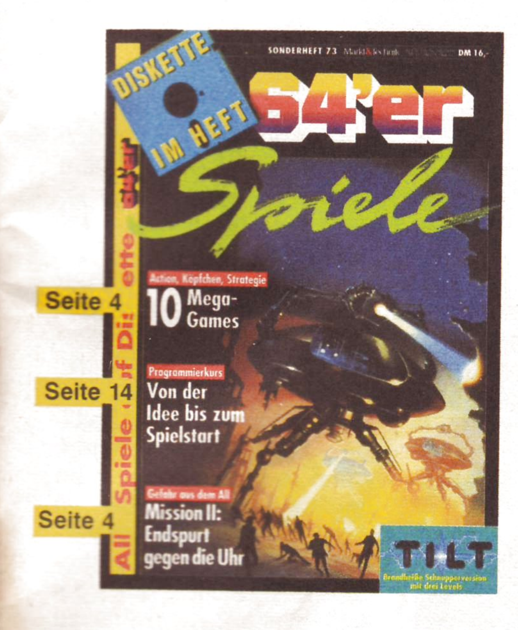
Machen Sie Musik mit Ihrem Amiga!
Mit den richtigen Hardware-Voraussetzungen wird Ihr Amiga zum Mini-Orchester. AMIGA-Magazin stellt Ihnen die interessantesten Sound-Sampler vor und zeigt Ihnen die Plus- und Minuspunkte.

AMIGA-Magazin sorgt für Ihren Wissens-Vorsprung: So sind Sie den anderen immer ein paar Bytes voraus...

Ab 18.12. im Handel!

Erst das AMIGA-Magazin macht Ihren Amiga perfekt!

INHALT





Kidnapper
Alles, was ein spannendes Abenteuerspiel ausmacht: Bei »Detektiv 2000« brauchen Sie jede Menge Grips und den Finger am Abzug (des Joysticks)!
Seite 42

Alle Programme zu Artikeln mit einem ■-Symbol finden Sie auf der beiliegenden Diskette (Seite 19).

Mission II – fieberhafte Bombensuche

TERROR AUS DEM ALL

Unter schwierigsten Bedingungen müssen gefährliche Space-Bomben entschärft werden – Auge in Auge mit der Gefahr. Nehmen Sie die Herausforderung an?

Feindliche Aliens aus den Tiefen des Andromeda-Nebels haben beschlossen, die Erde zu vernichten: 50 Ultra-Space-Bomben wurden mit programmiertem Kurs auf unseren Planeten abgefeuert. Glücklicherweise hatten die Außerirdischen aber eines nicht bedacht: Auf dem Weg zur Erde kreuzten die todbringenden Geschosse die Bahn des Asteroiden-Gürtels. Diese Gesteinsbrocken kreisen in der Nähe von Merkur und Pluto am Rande unseres Sonnensystems. Der Meteoritenschwarm bremste die Space-Bomben. Trotzdem ist die Gefahr groß, daß sich die eine oder andere Bombe von der steinernen Umklammerung löst und weiter auf die Erde zurast – was eine Katastrophe unbeschreiblichen Ausmaßes auslösen würde!



[1] Tolle Grafik, Supersound: Auf der Suche nach den gefährlichen Ultra-Space-Bomben.

Raumpatrouillen informieren den Obersten Rat der Weltregierung ständig über die aktuelle Situation, die von Tag zu Tag brenzliger wird: Es sieht so aus, als würden sich die Bomben aus dem Asteroiden-System befreien und ihre Bahn des Verderbens zur Erde fortsetzen.

Jetzt heißt es schnell handeln! Man bittet einen mutigen Haudegen und in vielen intergalaktischen Kriegen erprobten Kämpfer (nämlich Sie!), gegen hohe Belohnung zum Asteroiden-Gürtel zu rasen und die gefährlichen Bomben unschädlich zu machen.

Startbereit? Dann laden Sie das Spiel mit:

```
LOAD "MISSION II",8
```

und starten mit RUN. Zur Steuerung brauchen Sie den Joystick in Port 1.

Wenn das Titelbild (Abb. 1) erscheint, kommen Sie per <SPACE> oder Druck auf den Feuerknopf weiter. Die übrigen Programmteile werden nachgeladen: »by Blue Chip« und »Miss00«. Alle Grafiken des Programms wurden nach ausführlichem Test diverser Malprogramme schließlich mit »Amica Paint« entwickelt, da es sich am besten für dieses Vorhaben eignete. Unsere Tabelle zeigt einen Überblick der Speicheraufteilung.

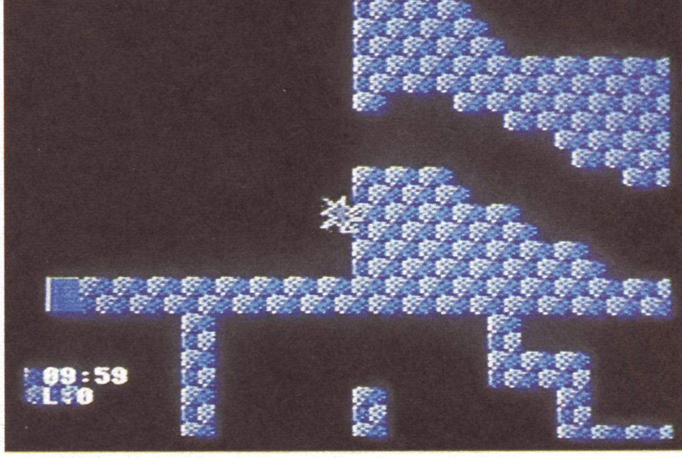
»Mission II« ist ein raffiniertes Geschicklichkeitsspiel. Unter echt miesen Bedingungen muß man in zehn Levels jeweils fünf Bomben entschärfen. Die Astroiden sind von riesigen Höhlen durchzogen (Abb. 2). Die natürliche Schwerkraft der überdimensionalen Gesteinsbrocken machen Ihrem Raumschiff erheblich zu schaffen. Nicht in allen Höhlen herrscht die gleiche Gravitation; außerdem darf man ein bestimmtes Zeitlimit nicht überschreiten. Wenn der Raumkreuzer an die Höhlenwände stößt, ist er futsch. Zu Beginn des ersten Levels hat man fünf Leben, das Zeitlimit fängt bei 9:59 min an, rückwärts zu zählen.

Der Bildschirm scrollt in vier Richtungen. Sie sehen aber nur immer einen Ausschnitt des aktuellen Levels: Auf Anhieb läßt sich nicht erkennen, wo die Bomben versteckt sind!

Wer es schafft, vor Ablauf des Zeitlimits alle fünf Bomben in der Gesteinshöhle einzusammeln, erhält einen Zeitbonus gutgeschrieben. Dann geht's im nächsten Level weiter. Außerdem erhöht jede entschärfte Bombe das Punktekonto.

Kurzinfo: Mission II

Programmart: Geschicklichkeitsspiel
Laden: LOAD "MISSION II",8
Starten: nach dem Laden RUN eingeben
Steuerung: Joystick Port 1
Besonderheiten: Spiel besitzt zehn Levels. »Trainer.Cde« installiert den Cheat-Modus.
Benötigte Blocks: 254
Programmautor: Andreas Montecchio



[2] Hüten Sie sich davor, die Höhlenwände mit Ihrem Raumschiff zu berühren!

Cheat-Modus inbegriffen

Die Datei »Trainer.Cde«. Sie hilft allen Spielern, mit den verzwickten Schwerkraft-Verhältnissen in den einzelnen Levels zurechtzukommen. Achtung: Ihr Computer muß dazu allerdings einen Reset-Schalter besitzen (oder ein entsprechendes Modul, in dem so ein Schalter eingebaut ist).

1. Laden und starten Sie das Spiel wie beschrieben.
2. Wenn das Intro erscheint, müssen Sie den Reset-Taster drücken.
3. Jetzt laden Sie den Cheat-Modus:

```
LOAD "TRAINER.CDE",8,1
```

4. Anschließend kann das Spiel mit »SYS 3968« erneut gestartet werden: Es lädt wieder die Datei »Miss00«. Es kann passieren, daß die Logos beim Nachladen des Levels durch die Systemmeldungen.

(Andreas Montecchio/bl)

Lustige Diamantenhatz

ROLLING STONES

»Roll it!« ist die ultimative Herausforderung für Joystick-Akrobaten. Achtung: Wir übernehmen keine Haftung für verrenkte Finger oder Daumen!

Das Spiel verfolgt ein neues Prinzip, bei dem es nicht nur auf Geschicklichkeit, sondern vor allem auf die richtige Taktik ankommt (quasi ein »Grübel-Geschicklichkeits«-Spiel): Sie müssen einen blaugetupften Ball und einen Edelkristall zum Ausgang des jeweiligen Levels rollen und dabei jede Menge Diamanten horten.

Laden Sie das gepackte Programme mit:

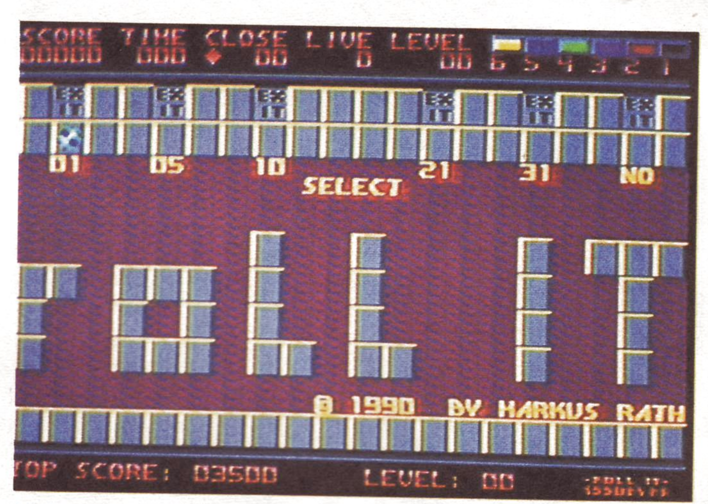
```
LOAD "ROLL IT!",8
```

und starten Sie mit RUN.

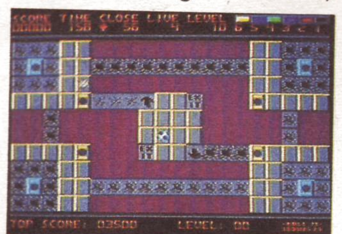
Nach dem Entpacken erscheint das Titelbild (Abb. 1). Die obere Bildschirmzeile bringt die Anzeigen für Score, Time, Close, Lives und Level. Der Spielball befindet sich in Level 1. Um diese Spielstufe zu aktivieren, müssen Sie den Ball ins Feld »Exit« bewegen. Es gibt aber noch die Möglichkeit, mit den Levels 5, 10 (Abb. 2), 21 oder 31 zu beginnen. Das Feld »Exit« bei »No« dient zum Verlassen des Spiels – aber nur, wenn das Programm diesen Entschluß akzeptiert!

Kurzinfo: Roll it!

Programmautor: Geschicklichkeit
Laden: LOAD "ROLL IT!",8
Starten: nach dem Laden RUN eingeben
Steuerung: Joystick Port 2
Besonderheiten: Alle Levels besitzen ein Zeitlimit
Benötigte Blocks: 40
Programmautor: Markus Rath



[1] Im Titelbild wählt man aus, in welchem Level gestartet wird. Wir geben Ihnen Brief und Siegel: Keiner ist leicht zu überstehen!



[2] Der zehnte Level des Spiels ist gespickt mit Schwierigkeiten

Steuerung

Der Ball läßt sich mit dem Joystick Port 2 in alle vier Richtungen bewegen. Den Kristall können Sie aber nur bewegen, wenn Sie den Ball auf einem der Richtungspfeile plazieren und den Feuerknopf drücken. Dann rollt der Edelkristall um ein Feld in der gewählten Richtung weiter.

Richtungspfeile

Sie haben nur eine begrenzte Funktionsdauer. Das läßt sich aus der Farbpalette in der oberen Bildschirmanzeige ablesen: Die Zahlen darunter bezeichnen die Felder, um die der Kristall bewegt werden kann. Achtung: Rollt er auf eine Pfeilplattform, wird der blaue Ball in die Richtung des Pfeils abgelenkt und kann im Nirwana landen.

Tödliche Felder

Roter Wüstensand (Energiemangel) und blauer, pulsierender Nährboden (High Energy) dürfen weder vom Ball noch vom Kristall berührt werden. Alle grauen Plattformen, egal ob Schalter, Ausgang (Exit) usw., lassen sich dagegen gefahrlos betreten. Vorsicht: Sie sind sehr brüchig! Je nach Grundzustand kann man sie bis zu dreimal überrollen, dann lösen sie sich auf. Das gilt aber nur für den Ball: Der Kristall darf so oft darüberfahren, wie er will.

Ausgänge

Die Level-Ausgänge (Exit) sind nur dann offen, wenn Sie eine bestimmte Anzahl Diamanten eingesammelt haben. Diamanten, die mit dem Energiekristall aktiviert und eingesammelt wurden, zählen doppelt. Beispiel: Brauchen Sie noch zwei Diamanten, um in den nächsten Level zu kommen, reicht es, wenn Sie mit dem Kristall nur einen einzigen aufnehmen. Es kommt zwar selten vor, aber manche Levels besitzen zwei Ausgänge. Jetzt liegt die Entscheidung allein beim Spieler, für welchen er sich entscheidet. Dabei kann es allerdings passieren, daß man einen Level zurückgesetzt wird.

Punkte

Für jeden Diamanten gibt's 15 (per Ball gesammelt) oder 30 Punkte (aufgenommen mit dem Edelkristall).

Schalter

Diese wichtigen Spielelemente lassen sich nur mit dem Kristall aktivieren. Jeder der drei verschiedenen Schalter (türkis, weiß, gelb) erfüllt zwei Funktionen:

- Option 1 (türkis): Aus dem Nährboden wachsen neue Diamanten.
- Option 1 (gelb): Es bilden sich brüchige Plattformen dritten Grades.
- Option 1 (weiß): Grüne Schalter entstehen.
- Option 2 (türkis): Kristall und Ball tauschen die Positionen.
- Option 2 (gelb): Jetzt läßt sich auch der Edelkristall zumindest 30 s lang mit dem Joystick steuern.
- Option 2 (weiß): erhöht alle Pfeile um einen Energiewert.

Nährboden

Wenn man brüchige Plattformen zerstört, erzeugt man Nährboden für Diamanten. Ebenso hinterläßt jeder eingesammelte Diamant diese Nährfläche (blauumrandete Plattform).

Hinweise zum Spiel

Jeder Level hat ein Zeitlimit, in dem Sie die Aufgabe bewältigen müssen. Schaffen Sie es nicht, verlieren Sie eines der insgesamt vier Leben. Wird der Level erfolgreich abgeschlossen, schreibt das Programm die verbleibende Zeit Ihrem Punktekonto gut. Wenn Sie einen neuen Tages-Highscore erreichen, wird der Continue-Modus aktiviert. Auch nach Verlust aller Leben können Sie dann in dem Level weitermachen, den Sie bis dahin erreicht hatten. Sehr oft kommt es vor, daß der Ball in einer ausweglosen Situation festsitzt: Dann bricht man diese Spielrunde mit der Taste < + > ab. Das gehortete Punktekonto wird dabei aber gelöscht: Ohne Pardon muß man wieder ganz am Anfang beginnen.

Bei den vielen Fallen, die der Autor eingebaut hat, ist es nicht so leicht, alle 40 Levels auf Anhieb zu schaffen: Langer Spielespaß ist vorprogrammiert!

(Markus Rath/bl)

Eruca – kleiner Wurm auf Wanderschaft

Friß oder kriech!

Vor ihm ist nichts sicher: »Eruca«, der Regenwurm, bohrt sich durch Äpfel, Birnen, Kirschen und leckere Eiswaffeln. Paradiesisch – wenn man nicht selbst so appetitlich wäre. Begleiten Sie ihn durch den gefährlichen Stadtpark!

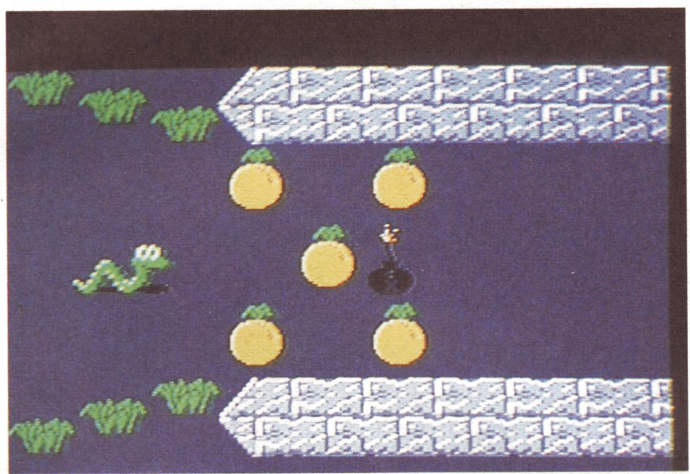
Was gibt's Langweiligeres, als ständig am selben Apfel zu knabbern? Eruca, hoffnungsvoller Sprößling aus dem Stamm der Westpark-Würmer, hat es endgültig satt: Er möchte die Welt kennenlernen und macht sich auf den Weg zum East End.

Wenn Sie den Kriechling begleiten möchten, müssen Sie das Spiel mit:

```
LOAD "ERUCA",8
```

laden und anschließend RUN eingeben.

Nach dem Entpacken erscheint der Titel-Screen mit Hinweisen über die Programmautoren und den aktuellen High score. Per Feuerknopf (Joystick Port 2) kommen Sie in den ersten Level (Abb. 1). Der untere Bildschirmteil zeigt einen inversen Balken (Energy), den Sie ständig im Auge behalten sollten. Der Wurm muß nun versuchen, alles zu fressen, was ihm in den Weg kommt: Grasbüschel, reife und grüne Äpfel, saftige Kirschen und blaue Pflaumen. Damit erhöht sich sein Punktekonto (Score) auch der Energiebalken nimmt nicht so schnell ab.



[1] Unterwegs findet der Wurm jede Menge zu futtern

sich in den High Score eintragen (Abb. 3). Voraussetzung: Sie müssen mindesten 2001 Punkte erreichen. Tippen Sie Ihren Namen ein und drücken Sie anschließend <RETURN>. Die Bestenliste gilt aber nur fürs aktuelle Spiel und wird nicht auf Diskette gespeichert. Wenn der Vorspann wieder erscheint, kommt man per Feuerknopf erneut in Level 1.

Verschiedene Spielstufen

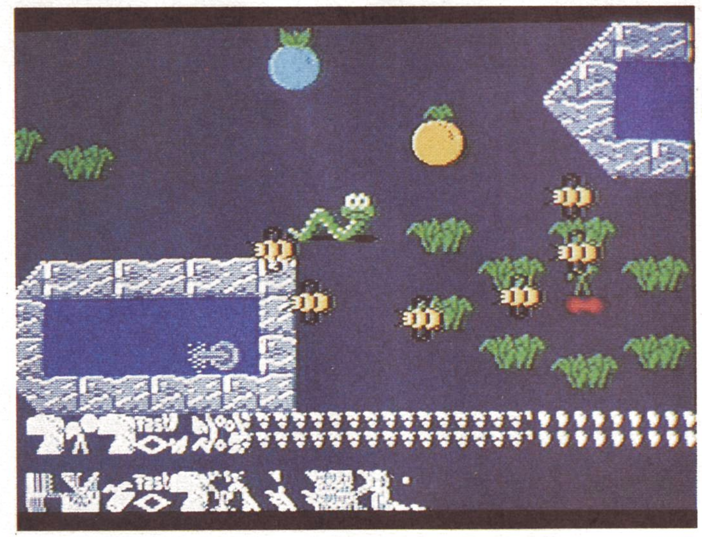
Der Bildschirm scrollt nach links und enthüllt die Spiele-Landschaft. Das Spiel besitzt vier Levels, pro Spielstufe werden ca. 25,5 Bild-

ERUCA

HI - SCORE

12000 GRAPHMACHINE
06000 MARKT&TECHNIK
03000 GRAPHMACHINE
02602 SYLVIA_
02000 MARKT&TECHNIK

[3] Bis man in den High score kommt, muß der Wurm viele Äpfel fressen...



[2] Bienenschwärme knabbern am Energievorrat

Kurzinfo: Eruca

Programmart: Geschicklichkeit & Action
Laden: LOAD "ERUCA",8
Starten: nach dem Laden RUN eingeben
Steuerung: Joystick Port 2
Besonderheiten: Vier Levels mit jeweils 25 Bildschirmen
Benötigte Blocks: 62
Programmautoren: Antje Keller/Oliver Rauscher

Viele Feinde bedrohen Eruca jedoch im unbekannten Gelände: Schlangen, Bienen, Käfer und anderes Getier. Außerdem muß sich der Wurm vor Mauern, Bomben mit brennender Lunte, Stacheldraht und Messern hüten (Abb. 2). Wenn man mit solchen Widrigkeiten in Berührung kommt, ist das zwar nicht lebensgefährlich, zehrt aber am Energievorrat. Bei solchen Kollisionen macht das Würmchen ein recht trauriges Gesicht, als hätte es einen wurmstichigen Apfel erwischt...

Zum Glück verstecken sich auf der Wiese des großen Parks auch nützliche Extras:

- »Fast« beschleunigt die Steuerung des Wurms,
- »Ghost« macht Eruca unverwundbar.

Beide Sonderfunktionen gelten allerdings nur einige Sekunden – dann ist alles wieder wie gehabt. Im »Ghost«-Zustand lassen sich nur Punkte sammeln, der Energievorrat verringert sich aber nicht.

Auch, wenn man nicht alle Levels schafft und vorzeitig wegen Energiemangel aussteigen muß, darf man schirmseiten verwendet. Haben Sie einen Level erfolgreich gelöst, erreichen Sie automatisch den nächsthöheren. Grafik und Zeichensatz zu diesem Spiel entstanden mit einem C-128-Programm: »MSG« im 128'er-Sonderheft 22. Sound und Animation der Bienen- und Käferschwärme wurden von den Autoren mit eigenen Programmen entwickelt.

Weit ist der Weg nach Osten! Hoffen wir, daß unser Wurm unterwegs viel Futter findet und unbeschadet auf der anderen Seite des Parks landet. (Antje Keller/bl)

Tilt – Schnupper-Version mit drei Levels

Bally hoo!

Beim Amiga schon fast ein Oldie: »Tilt«, das verzwickte Geschicklichkeitsspiel für eine Metallkugel und viele Labyrinthe gibt's jetzt auch für den C64. Wir haben ein Vorab-Game für Sie aufgetrieben. Die drei Level unserer Demoversion machen Appetit aufs Originalspiel!

Es ist schon eine sonderbare Welt, in der die kleine Metallkugel Bally lebt: Ein unübersichtliches Höhlenlabyrinth aus metallisch glitzernden Schienen, zwei bodenlosen, schwarzen Löchern, Stop-Schildern und vielen Fragezeichen. Bally schnellt aus dem Startloch und saust im Röhrensystem herum. Der Rundling sammelt Punkte, Sonder-Boni und versucht dabei, das Ziel möglichst schnell zu erreichen. Eine feindliche Macht, die Atair-Androiden, haben den kleinen Metallkugeln den Kampf angesagt und möchten sie vernichten. Natürlich haben diese Maschinenungeheuer auch Bally aufs Korn genommen und versuchen, der Kugel mit Hindernissen und hinterhältigen Fallen den Garaus zu machen. Helfen Sie Bally, so schnell wie möglich, alle Hindernisse zu überwinden und das Ziel sicher zu erreichen.

Laden Sie die Demoversion mit:

```
LOAD "TILT /TC",8
```

und starten sie mit RUN.

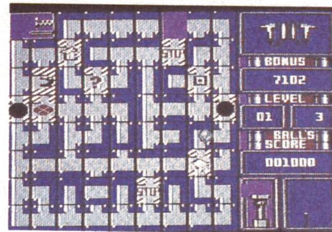
Nach dem Entpacken erscheint das Spielmenü (Abb. 1). Die einzelnen Funktionen werden per Joystick Port 2 gewählt:

- Ein- oder Zwei-Spieler-Modus,
- Hall of Fame (High score: mindestens 10000 Punkte müssen erreicht werden),
- Difficulty (Schwierigkeitsgrad, 1 = leicht bis 9 = megaschwer),
- Musik an/aus (damit kann man die Sound-Begleitung während des Spiels abschalten).

Per Feuerknopf kommt man ins Spiel (Abb. 2). Oben links katapultiert eine Metallfeder die Kugel ins Spielfeld (Level 1). Der Ausgang ist im Loch auf der rechten Seite. Mit dem Joystick muß man die Einzelteile des Spielfelds so verschieben, daß eine durchgehende Strecke entsteht, auf der Bally sicher das Ziel erreicht. Dafür gibt's selbstverständlich nicht nur einen Lösungsweg, sondern viele Möglichkeiten, die Level-Bausteine zu ordnen. In kritischen Situationen läßt sich die Metallkugel durch Joystick-Ruck in die gewünschte Richtung und gleichzeitigem Druck auf den Feuerknopf aus der Gefahrenzone befördern.



[1] Zuerst muß man den Spielmodus und den Schwierigkeitsgrad einstellen

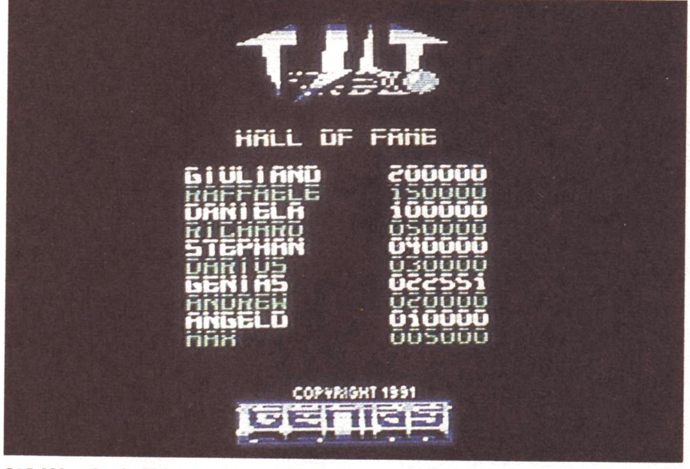


[2] Wer soll sich denn hier noch auskennen?

Im Zwei-Spieler-Modus gelten andere Bedingungen: Die Level-Felder bewegt man per Joystick, die Kugel aber mit der Tastatur:

<**9**>: Kugel nach links,
<**0**>: nach rechts,
<**$**>: aufwärts,
<*****>: nach unten,
<**RUN/STOP**>: Pausentaste. In der Anzeigenleiste am rechten Bildschirmrand erscheint die Meldung: Frozen. Mit erneutem Druck auf den Feuerknopf geht's weiter.

<**Pfeil links**>: bricht das Spiel ab. Die Bestenliste erscheint, erneuter Druck auf den Feuerknopf bringt Sie wieder ins Spielmenü.



[3] Wer bei »Tilt« in den High Score kommt, hat gefährliche Situationen gemeistert – oder viele Extras gesammelt!

Jede Spielrunde beginnt mit drei Kugeln (Leben) und einem Bonus von 8500 Punkten, der sich aber wahnsinnig schnell verringert, je nach eingestelltem Schwierigkeitsgrad. Im Labyrinth trifft man auf farbige Extras, die bei Berührung Balsam fürs Punktekonto sind:

- blaues Objekt: 400 Zeiteinheiten zusätzlich,
- rot: 1000 Punkte,
- lila: ein Extraleben.

Wie gemein die Fallen und Hindernisse sind, ist ebenfalls vom gewählten Schwierigkeitsgrad abhängig. Wenn alle Metallkugeln vorzeitig verbraucht sind, ist die Runde verloren.

Stößt die Kugel aufs Feld mit dem Fragezeichen, läuft ein Sprite-Männchen von links über den Bildschirm und macht sich an der Anzeige für die Restleben zu schaffen: Meist hat man danach eines weniger. Geraten Sie auf ein Stop-Feld, ist die Kugel einige Sekunden bewegungslos: wertvoller Zeitgewinn, um Spielelemente zu ordnen.

Wer mit seiner Kugel genügend Punkte hamstert (mehr als 10000), kann sich im High score (Abb. 3) verewigen.

Wir wünschen viel Spaß mit dieser abgespeckten Fassung, die aber alles bietet, was man von einem guten Geschicklichkeitsspiel erwartet. Unserer besonderer Dank gilt der Fa. Linel in der Schweiz, die uns die Demo-Version von »Tilt« für unsere Sonderheft-Diskette zur Verfügung gestellt hat. Wenn Sie Lust auf die umfangreichere Originalfassung bekommen haben: Die gibt's in den Computer-Abteilungen der Kaufhäuser und im Fachhandel (Vertrieb: Rushware). (bl)



Spiele bis zum Schlußbild zu lösen – schöner als Weihnachten! Unsere Tips & Tricks zu heißen Games machen die Tür zum High score weit auf.

Was, Sie wollen Ihren C64 auf dem nächsten Flohmarkt zum Schleuderpreis verkaufen? Nur, weil Sie schon wieder im 65. Level des verzwickten Action-Games hängengeblieben sind? Nur die Ruhe: Wir liefern Ihnen jede Menge POKEs, SYS-Befehle, Cheat-Modi und Codewörter zu Spielen der 64'er-Sonderhefte und vielen kommerziellen Games. (bl)

64'er-Sonderheft 66 (Spiele)

KILLWARP vor dem Start mit RUN eingeben: POKE 2775,64. Damit wird die Sprite-Kollision abgeschaltet.

ATOM-X POKE 4458,0 bringt unendlich viel Zeit (Befehl vor dem Start eingeben!). Peter Senger in Lippstadt hat noch einen Cheat-Modus parat: Um ins jeweils nächste Level zu kommen, muß man die Tasten <A>, <T>, <O>, <M>, <X> und <SPACE> gleichzeitig drücken (Abb. 1).

CAVEMAN MIKE Aktivieren Sie nach dem Start den Reset-Taster. Jetzt müssen Sie POKE 49278,173 eingeben und mit SYS 49152 neu starten: Ab sofort sind Sie unsterblich resp. stehen Ihnen unendlich viele Leben zur Verfügung.

LASER-SCHACH vor dem Start mit RUN eingeben: POKE 4022,0: POKE 4027,0: POKE 4032,0. Damit ist Mogeln erlaubt.

MAGIC POKE 9123,169: POKE 9124,0: POKE 9125,234 (vor dem Start eingegeben) eliminiert alle Gegner.

SUBURBIA POKE 16342, 173 schaltet Sprite-Kollisionen ab.

PSYCHO Wir können Ihnen den richtigen Lösungsweg zu diesem verzwickten Abenteuerspiel anbieten. Entdeckt haben ihn – unabhängig voneinander – Tobias Eichler in Bühl und Stefan Müller in Reinbek:

```
S-S-OEFFNE SARG-NIMM AMULETT-W-W-W-W-NIMM BROT-
ESS BROT-W-NIMM PISTOLE-S-ZEIG AMULETT-DREH SCHILD
(Buchstaben notieren!)-N-N-N-NIMM SCHAEDEL-O-
SAGE (Buchstaben)-NIMM RING-O-NIMM KERZE-O-S-
ZEIG PISTOLE-NIMM FLOETE-S-GEH AUFZUG-DRUECK KNOPF-
GEH RAUS-SPIEL FLOETE-GEHE HOEHLE-WIRF RING-
SPALTE SCHAEDEL
```

64'er-Sonderheft 60 (Adventures)

Die Teillösung zum Adventure »Grab des Pharao« (Abb. 2) wird Sie ein ganz schönes Stück weiterbringen. Sie stammt von Steffen Woischke.

```
GEHE O-NIMM SPITZHACKE-GEHE N-FRAGE BEDUINE-GEHE O-
GEHE N-HACKE STRAUCH-ZERSTOERE EINGANG-
KRIECHE DURCH-GEHE W-GEHEW-NIMM STATUE-ZIEHE HEBEL-
VERLIERE STATUE-GEHE W-GEHE N-NIMM SEIL-GEHE O-
ZERSTOERE WAND-KRIECHE GEHEIMGANG-GEHE N-GEHE N-
ZERSTOERE PFEIL-OEFFNE TUER-GEHE N
```

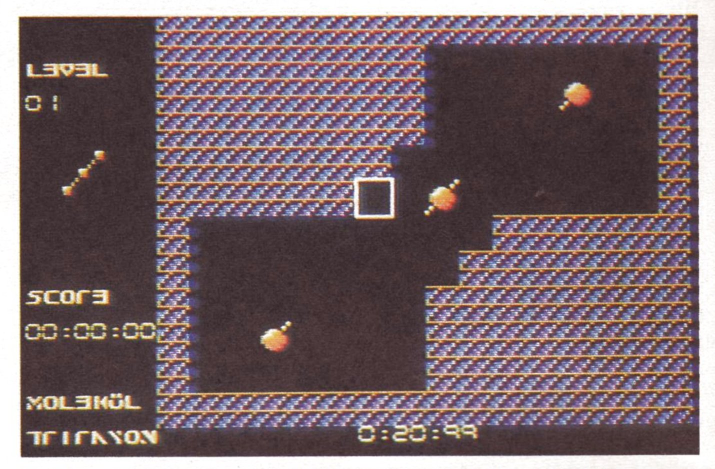
Noch ein Tip: Klopfen Sie an die Wand im Osten!

Unsere gallischen Freunde »Asterix und Obelix« hat Helga Schlicht einige Meilen auf ihrer Odyssee begleitet:

```
NIMM SICHEL-NIMM SCHWERT-O-KLOPFE TUER-GIB SICHEL-
O-NW-NIMM ZETTEL-LIES ZETTEL-KITZEL SCHLANGE-
NIMM GOLD-SO-N-TRINKE ZAUBERTRANK-TOETE ROEMER-
KLETTERE IN SCHIFF-N-NIMM SEIL-S-W-W-
KLETTERE LEITER
-NIMM KEULE-W-KAUFE KLEIDER-JA-TRAGE KLEIDER-O-
VERKNOTE PATROUILLE-BERUEHRE PATROUILLE-SCHAU-
ERKUNDE-ENTFERNE PATROUILLE-GIB SEIL-GIB SCHWERT-
VERPRUEGLE MIT KEULE
```

64'er-Magazin, Ausgabe 2/91

»Puzzlenoid«: Verschieben Sie bei Spielbeginn die Steine links oben eine Stelle nach rechts und den oberen rechten



[1] Atom-X: Eine fingerbrecherische Tastenkombination aktiviert den Cheat-Modus



[2] Grab des Pharao: Der wichtigste Hinweis kommt vom Mann im weißen Burnus.

Stein nach links. Jetzt kommt der Quader links unterhalb des zuletzt verschobenen Steins dran. Der Rest ist dann nur noch ein Kinderspiel.

64'er-Magazin, Ausgabe 4/91

Zum Spiel »Future Dungeons« hat Nicki Heusler die Zugangcodes für alle 26 Level (0 bis 25) ausgetüftelt (Tabelle 1). Nach einem Reset kann man das Spiel mit SYS 20486 neu starten.

Die folgenden Spieletips und Cheat-Modi stammen aus »64'er« und »Power Play«:

CHIP'S CHALLENGE Die Levelcodes zu diesem Geschicklichkeitsspiel sind geknackt! Man findet sie in Tabelle 2.

GRAND PRIX CIRCUIT Wenn Sie auf der Rennstrecke »Silverstone« durch die Box fahren, läßt sich der Bolid problemlos durch die kommende Kurve steuern (Abb. 3).

KOLONIALMACHT Will man hohe Preise durchsetzen, muß man stets zehn Tonnen stehen lassen und Kriege vermeiden. Möchten Sie Ihre Stadt gegen Feinde sichern, sind eine große Armee oder viele Plantagen nötig. Legen Sie einen Plantagen-Ring rund um die Stadt, halten Sie dabei aber ausreichend Abstand. Nutzen Sie auch das Gelände (Flüsse, Gebirge). Kleine Lücken lassen sich nämlich mit nur wenigen Soldaten verteidigen.

MANIX Haben Sie auch eine Schwäche für Gummibälle? Hier sind die Codes der 15 Levels:

MANIX (1), ZONE (2), SPACE (3), MOON (4), TIME (5), MOTIVATE (6), TOM (7), MAJOR (8), MIKE (9), SARAH (10), DOUGH (11), NEIL (12), IXION (13), KINETIC (14), TRAPCLIMAX CLIMAX (15).

MARS SAGA Die Überlebenschancen hängen von den ausgewogenen Skills der Partymitglieder ab. Es hat sich herausgestellt, daß vier Personen ausreichen, um das Spiel erfolgreich zu beenden. Am besten nehmen Sie neben Tom Jetland einen »Marine« mit, denn der beherrscht den »Auto«-Skill. Ein Mediziner mit geschickten Händen und Erfahrung sollte ebenfalls dabei sein. Außerdem wäre es vorteilhaft, wenn der Doktor programmieren kann. Die vierte Figur ist nicht so wichtig, sie kann z.B. mit einem Polizisten besetzt werden.

Je ein Partymitglied sollte man auf den »Rifle« bzw. »Auto«-Skill schulen und diese Fähigkeiten während des Spiels ausbauen. Der Doc sollte seinen »Medic«-, vor allem aber seinen »Program«-Skill erweitern. »Throwing« ist äußerst nützlich, denn es läßt Waffen mit flächendeckender Wirkung zu (z.B. Molotow-Cocktails). Hohe »Program«-Skills öffnen Tür und Tor. Während des Hackens am Terminal erhält man den Minenzugangs-Code und die Erlaubnis, den »War Games Room« zu betreten. Zusätzlich löscht man so sein Strafregister in den Polizeiakten oder besorgt sich einen Paß, mit dem man das »Speeder Transport«-System gratis benutzen darf. Das letzte Programm verschafft schließlich den wichtigen »Golum«-Skill.

Phelos Fletcher, den man suchen muß, hält sich im Mineneingang von Progeny versteckt.

Im Munitionsshop treffen Sie Cybil Graves, die einen Auftrag für Sie hat. Dazu muß man die Nomaden aufsuchen. Von ihnen erhalten Sie einen Peilsender (Finder), mit dem man ein Päckchen für Cybil suchen muß. Sie finden es in südöstlicher Richtung, nahe bei Primus. Ist man an der richtigen Stelle angelangt, hört der Peilsender auf, zu blinken und zeigt konstantes Licht. Um an das Paket zu kommen, muß ein Party-Mitglied im Besitz eines »Mining Lasers« sein. Den entdeckt man in der Mine von Primus: Dort liegen genügend herum. Dann liefert man das Päckchen bei Cybil ab und kassiert dafür eine Belohnung.

An der Stelle, an der vorher das Päckchen lag, ist nun ein »Volcanic Vent« aufgetaucht. Es führt in ein Höhlensystem. Durchstöbern Sie diesen Ort ausgiebig, denn die Marsbewohner haben dort allerlei nützliche Dinge vergessen (z.B. das »Reaver Rifle«, das beste Gewehr des Spiels).

Falls Cybil polizeilich gesucht wird, findet man sie im Armour Shop von Parallax.

[3] Grand Prix Circuit: schnelle Flitzer und dröhnende Motoren

Die Repair Shops in Progeny liefern wichtige Informationen, wenn man »Clerk Credits« anbietet. Besucht man anschließend das Kampfcenter in Parallax, kann man »Golum« trainieren. In der Spielhalle von Parallax bekommt man die »Golum Armour«.

Es lohnt sich auf alle Fälle, die Universität von Parallax zu besuchen. Dort kann mich sich den Pfeilsender so justieren lassen, daß er auf einen weiteren »Volcanic Vent« weist. Er führt über drei Etagen nach Proscenium.

MR. HELI Codefolgen für höhere Levels:
- Level 2: CAAFDHCAAUJJJJJDCKCY
- Level 3: DAANHEADAAUJJJJJDCKCZ

NEW ZEALAND STORY Unendlich viele Leben gibt's mit POKE 3215,173.

NORTH SEA INFERNO Ist man von Gegnern umzingelt, ballert man sich eine Seite frei und gibt Fersengeld. Wenn Sie dann wieder zurückkehren, sind die restlichen Feinde verschwunden.

OIL IMPERIUM Im Büro muß man folgende drei Tasten gleichzeitig drücken: <Pfeil links>, <CTRL> und <RUN/STOP>. Dann erhöht sich Ihr Guthaben von fünf auf 100 Millionen Dollar! Am Monatsende erscheint auf dem unteren Bildschirmrand der Hinweis, daß Ihnen eine Tante dieses hübsche Sümmchen vererbt hat (Abb. 4).

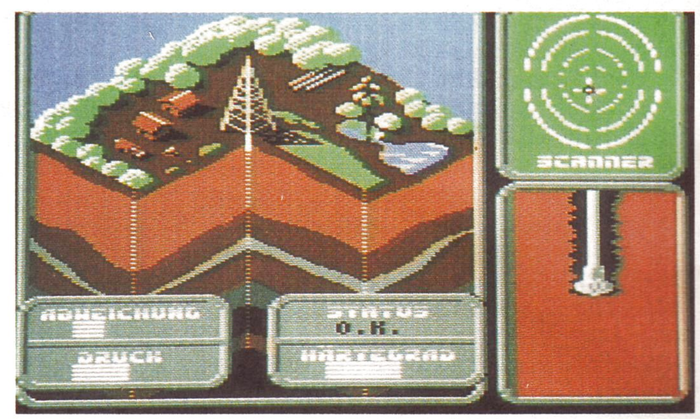
PAC MANIA Reset durchführen. POKE 28520,165 ergibt unendlich viele Leben, POKE 22459,137 schaltet die Sprite-Kollisionen ab. Mit SYS 14336 kann man neu starten.

Levelcodes zu »Future Dungeons«

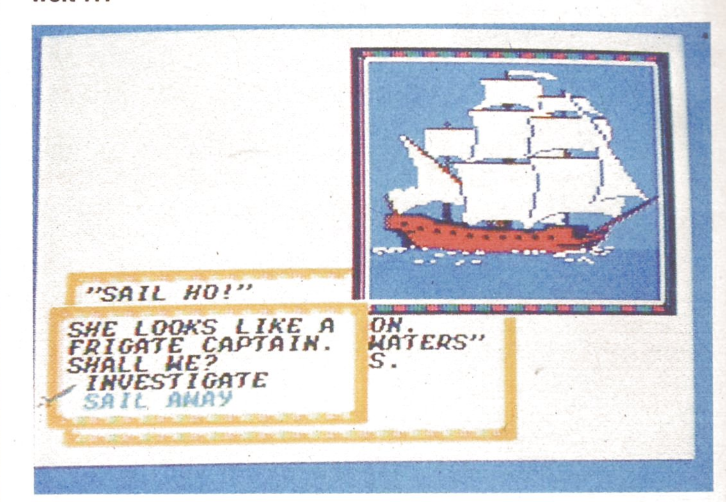
Level	Code
Level 1	010000000
Level 2	199120011
Level 3	187653210
Level 4	909090909
Level 5	331100009
Level 6	299188966
Level 7	248350972
Level 8	519437003
Level 9	981276450
Level 10	125863119
Level 11	999111000
Level 12	872117512
Level 13	987123650
Level 14	001232100
Level 15	110190000
Level 16	009871230
Level 17	010101010
Level 18	280000285
Level 19	990099000
Level 20	127365728
Level 21	097219752
Level 22	872112785
Level 23	962026092
Level 24	222222221
Level 25	097219620

Levelcodes zu »Chip's Challenge«

Level	Name	Code
1	Lesson 1	BDHP
2	Lesson 2	JXMJ
3	Lesson 3	ECBQ
4	Lesson 4	YMCJ
5	Lesson 5	TQKB
6	Lesson 6	WNLP
7	Lesson 7	FXQO
8	Lesson 8	NHAG
9	Nuts and Bolt	KCRE
10	Brush Fire	VUWS
11	Trinity	CNPE
12	Hunt	WVHI
13	Southpole	OCKS
14	Teleblock	BTDY
15	Elementary	COZQ
16	Cellblocked	SKKK
17	Nice Day	AJMG
18	Castle Moat	HJML
19	Digger	MRHR
20	Tossed Salad	KGFP
21	Iceberg	UGRW
22	Forced Entry	WZIN
23	Blobnet	HUVE
24	Orto Geld	UNIZ
25	Blink	PQGV
26	Chchchips	YVYJ
27	Go with the flow	IGGZ
28	Ping Pong	UJDD
29	Articflow	QGOL
30	Mishmesk	BGZP
31	Knot	RYMS
32	Scavenger Hunt	PEFS
33	On the rocks	BQSN
34	Cypher	NQFI
35	Lemmings	VDTM
36	Ladder	NXIS
37	Seeing stars	VQNK
38	Sampler	BIFA
39	Glut	ICXY
40	Floor gas borg	YWFH
41	I.C.you	GKWD
42	Beware of bug	LMFU
43	Lock block	UJDP
44	Refraction	TXHL
45	Monster Lab	OVPZ
46	Three dors	HDQV
47	Pier seven	LXPP
48	Mugger square	JYSF
49	Problems	PPXI
50	Digdirt	QBDH
51	Islide	IGGJ
52	The last laugh	PPHT
53	Traffic cop	CGNX
54	Grail	ZMGC
55	Potpoum	SJES
56	Deepfreeze	FCJE
57	Strange maze	UBXU
58	Loop around	YBLT
59	Hidden danger	BLDM
60	Scoundrel	ZYVI
61	Rink	RMOW
62	Slomo	TIGW
63	Factory	GOHX
64	Spooks	IJPQ
65	Amsterdam	UPUN
66	Victim	ZIKZ
67	Chipmine	GGJA
68	Eeny miny moe	RTDI
69	Bounce city	NLLY
70	Nightmare	GCCG



[4] Oil Imperium: J. R. Ewing läßt grüßen, doch Dallas ist weit ...



[5] Pirates: pralle Segel und 'ne Buddel voll Rum!

PIRATES So erhält man eine schlagkräftige Mannschaft: Man fährt in eine Stadt, drückt <RUN/STOP> und startet erneut mit RUN. Jetzt besucht man die nächste Kneipe und heuert einige Matrosen an. Das kann man so oft wiederholen, bis die eigenen Schiffe voll bemannt sind (Abb. 5).

PRO BOXING SIMULATOR Die Paßwörter für alle Boxer:
- Steady Eddie: PARTY,
- Dirty Larry: TALON,
- Fast Freddie: SWORD,
- Ronnie Razor: LUCKY,
- Deadly Dan: UNION.

RAINBOW ISLANDS Die verschiedenfarbigen Diamanten sollte man in der richtigen Reihenfolge aufsammeln (von links nach rechts). Dann erreicht man eine Schatzkammer und nimmt ein Extra mit, das einem immer erhalten bleibt – auch wenn man ein Leben verliert.

REALM OF THE TROLLS Reset auslösen. POKE 23889,173 bringt unendlich viele Leben, POKE 23948,173 ergibt unendlich viel »Mental«, POKE 24113,173 hält den Timer an. Durch die Eingabe von SYS 34389 kann man neu starten.

ROBOCOP Nach dem Programmstart muß man den Reset-Knopf drücken. Mit folgenden POKEs bekommen Sie Probleme des Zeitlimits und des Energievorrats voll in den Griff: POKE 44179,96 (Zeit), POKE 44368,96 und POKE 44392,96 (Energie). Wenn das erste Mal nachgeladen wurde, muß das Spiel erneut mit einem Reset unterbrochen werden, dann POKE 45166,96 (Zeit), POKE 45348,96 und POKE 45324,96 (Energie) eingeben. Für den dritten Level gelten folgende Befehle: POKE 42960,96 (Zeit), POKE 43131,96 und POKE 43155,96 (Energie). Beim Start des dritten Levels kann es passieren, daß eine wirre Grafik erscheint. Dann drückt man <RUN/STOP> und startet das Level per Feuerknopf neu. Alle drei Levels lassen sich mit SYS 32768 starten.

SDI Nach gleichzeitigem Druck auf die Tasten <X>, <T>, <C> und <SPACE> kommt man in den Cheat-Modus.

SECRET OF THE SILVER BLADES Items verdoppeln: Gespeicherten Spielstand laden und in den Vault gehen. Jetzt Deposit wählen und die gewünschten Items »dropen«, anschließend Vault verlassen. Diese gespeicherte Situation erneut laden, zum Vault zurückkehren und die Gegenstände mitnehmen. Ab sofort besitzt man die Items doppelt, wie die Gegenstandsliste im Spiel beweist (Abb. 6).

SILKWORM Reset auslösen, POKE 4527,165 und erneut mit SYS 2080 starten

SIM CITY Legen Sie eine Stromleitung übers Wasser und entzünden Sie diese mit »Fire« im Desaster-Menü. Nach dem Planieren erwirbt man Land. Wenn Ihnen das Geld zu schnell ausgeht, müssen Sie öfters <F1> drücken.

SKI OR DIE Bei Aero-Aerials muß man den Joystick ruckartig nach links und rechts bewegen – das aktiviert einen Supersprung! Während der Schneeballschlacht ist es wichtiger, die Angreifer zu erwischen. Damit bleibt man recht lange im Spiel. Um die Bonuspunkte kann man sich immer noch kümmern.

SUPERSTAR ICE-HOCKEY Schnappen Sie sich den Puck und schießen aus einiger Entfernung kräftig aufs Tor. Ist die Scheibe in der Luft, muß man die RESTORE-Taste drücken. Damit's weitergeht, sollte man jetzt <SPACE> antippen. In den meisten Fällen landet der Puck im Tor.

THE DEEP Reset auslösen. POKE 8985,12: POKE 9067,12 stellt unendlich viele Schiffe zur Verfügung. Mit POKE 7509,12: POKE 14478,12 besitzt man unendlich viele Bomben, mit POKE 16036,7: POKE 16040,12 gibt's eine nie versiegende Anzahl Tauchglocken. Neu starten muß man mit zwei SYS-Anweisungen: SYS 16020: SYS 2064.

ULTIMA V Mit diesem Trick stehen allen Zauberern immer genügend Reagenzien zur Verfügung. Allerdings muß man schon mindestens zehn bis 15 Kräuter der verlangten Sorte besitzen. Dann beginnen Sie, die Reagenzien zu mischen. Auf die Frage, wie viele Sprüche gemischt werden sollen, muß man die Zahl »16« eingeben. Die Zauberformeln werden nun gemixt. Wenn man jetzt die Kräutervorräte überprüft, haben sie nicht abgenommen – im Gegenteil: Die Anzahl der soeben benutzten Kräuter ist auf 95 bis 99 gestiegen!

Ist die Party am Verhungern, sollte man die Empath Abbey aufsuchen. In der Kneipe liegen Lebensmittelpakete herum, die niemand gehören. Wer sie sich unter den Nagel reißt, die Leiter rauf- und wieder runterrennt, findet die Tische wieder neu gedeckt vor: Damit kann man seine Nahrungsvorräte preiswert aufstocken.

VENGEANCE Nach dem Laden Reset-Taster drücken. Mit POKE 20767,1 kann man die Sprite-Kollision abschalten. 255 Schiffe bekommen Sie mit POKE 20772,255 (obwohl auf dem Bildschirm nur die Zahl »9« erscheint). Unendlich viele Leben gibt's mit POKE 30083,234. Das Spiel wird mit SYS 18432 neu gestartet.

ZAK MC KRACKEN Wenn Zachary unerkannt in den Hinterraum der Telephone Company eindringen möchte, muß er nur den Hut und die Nasenbrille aufsetzen. Dann hält ihn der Außerirdische für seinen eigenen Bruder und läßt in durch die Tür hinter dem Tresen. Dort darf Zachary sogar die Maschinen bedienen und den Computer benutzen.

CASHCARD: Sie liegt unter Zacharys Schreibtisch. Mit dem Buttermesser kann man sie darunter hervorholen. Die entsprechende Befehlszusammenstellung lautet: Benutze Buttermesser mit Plastikkarte. Achtung: Wenn Zak das Einweckglas aufsetzt, erstickt er innerhalb von 30 Sekunden!

(Nicki Hensler/bl)

Levelcodes zu »Chip's Challenge«

Level	Name	Code
71	Corridor	LAJM
72	Reverse alley	EKFT
73	Morton	QCCR
74	Playtime	MKNH
75	Steam	MJDV
76	Four plex	NMRH
77	Invincible champion	FHIC
78	Force square	GRMO
79	Dawn and quatered	JINU
80	Vanishing act	EVUG
81	Writer block	SCWF
82	Socialist Action	LLIO
83	Up the block	OVPJ
84	Wars	UVED
85	Telenet	LEBX
86	Suicide	FLHH
87	City block	YJYS
88	Spirals	WZYU
89	Block busters	VCZO
90	Playhouse	OLLM
91	Jumping swarm	JPQG
92	Vortex	DTMI
93	Roadsign	REKF
94	Now you see it	EWCS
95	Four square	BIFQ
96	Paranoia	WVHY
97	Metastable to chaos	IOCS
98	Shrinking	TKWD
99	Catacombs	XUVU
100	Colony	QJXB



[6] Secret of the Silver Blades: Die Kampfstärke der Monster läßt sich vorher einstellen.



[7] Zak McKracken: Antike Stadt unter Wasser

64'er Sonderhefte

alle auf einen Blick

Die 64'er Sonderhefte bieten Ihnen umfassende Information in komprimierter Form zu speziellen Themen rund um die Commodore C 64 und C 128. Ausgaben, die eine Diskette enthalten, sind mit einem Diskettensymbol gekennzeichnet.

C 64, C 128, EINSTEIGER



SH 0022: C 128 III
Farbiges Scrolling im 80-Zeichen Modus / 8-Sekunden-Kopierprogramm



SH 0026: Rund um den C64
Der C64 verständlich für Alle mit ausführlichen Kursen



SH 0029: C 128
Starke Software für C 128/ C 128D / Alles über den neuen C 128D im Blechgehäuse



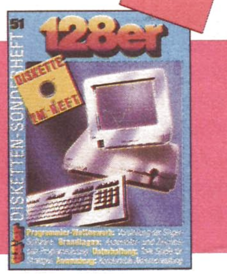

SH 0036: C 128
Power 128: Directory komfortabel organisieren / Haushaltsbuch: Finanzen im Griff / 3D-Landschaften auf dem Computer



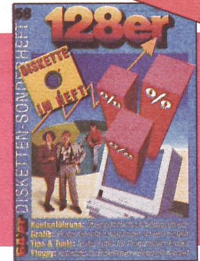
SH 0038: Einsteiger
Alles für den leichten Einstieg / Super Malprogramm / Tolles Spiel zum selbermachen / Mehr Spaß am Lernen




SH 0050: Starthilfe
Alles für den leichten Einstieg / Heiße Rythmen mit dem C 64 / Fantastisches Malprogramm



SH 0051: C 128
Volle Floppy-Power mit "Rubikon" / Aktienverwaltung mit "Börse 128"



SH 0058: 128er
Übersichtliche Buchhaltung zuhause / Professionelle Diagramme



SH 0062: Erste Schritte
RAM-Exos: Disketten superschnell geladen/ Exbasic Level II: über 70 neue Befehle/ Rafinessen mit der Tastatur

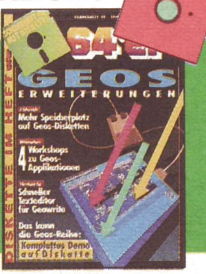


SH 0064: 128ER
Anwendungen: USA Journal / Grundlagen: CP/M, das dritte Betriebssystem/ VDC-Grafik: Vorhang auf für hohe Auflösung

GEOS, DATEIVERWALTUNG



SH 0028: Geos / Dateiverwaltung
Viele Kurse zu Geos / Tolle Geos-Programme zum Abtippen



SH 0048: GEOS
Mehr Speicherplatz auf Geos-Disketten / Schneller Texteditor für Geowrite / Komplettes Demo auf Diskette

SH 0059: GEOS
GeoBasic: Großer Programmierkurs mit vielen Tips & Tricks

PROGRAMMIERSPRACHEN



SH 0035: Assembler
Abgeschlossene Kurse für Anfänger und Fortgeschrittene



SH 0040: Basic
Basic Schritt für Schritt / Keine Chance für Fehler / Profi-Tools und viele Tips

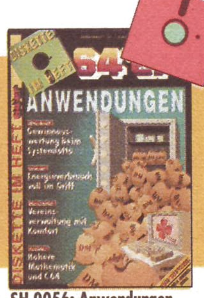
ANWENDUNGEN



SH 0031: DFÜ, Musik, Messen-Steuern-Regeln
Alles über DFÜ / BTX von A-Z / Grundlagen / Bauanleitungen



SH 0046: Anwendungen
Das erste Expertensystem für den C 64 / Bessere Noten in Chemie / Komfortable Dateiverwaltung



SH 0056: Anwendungen
Gewinnauswertung beim Systemlotto / Energieverbrauch voll im Griff / Höhere Mathematik und C64

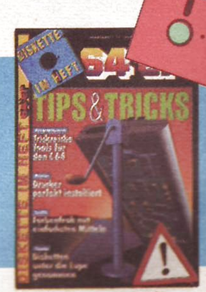
TIPS, TRICKS & TOOLS



SH 0024: Tips, Tricks& Tools
Die besten Peeks und Pokes sowie Utilities mit Pfiff



SH 0043: Tips, Tricks&Tools
Rasterinterrupts - nicht nur für Profis / Checksummer V3 und MSE / Programmierhilfen



SH 0057: Tips & Tricks
Trickreiche Tools für den C64 / Drucker perfekt installiert



SH 0065: Tips&Tools
Streifzug dirch die Zeropage/ Drucker-Basic: 58 neue Befehle zur Printer-Steuerung/ Multicolorgrafiken konvertieren/ über 60 heiße Tips&Tricks

HARDWARE



SH 0025: Floppylaufwerke
Wertvolle Tips und Informationen für Einsteiger und Fortgeschrittene



SH 0032: Floppylaufwerke und Drucker
Tips&Tools / RAM-Erweiterung des C64 / Druckerroutinen



SH 0047: Drucker, Tools
Hardcopies ohne Geheimnisse / Farbige Grafiken auf s/w-Druckern



SH 0067: Wetterstation:
Temperatur, Luftdruck und feuchte messen/ DCF-Funkuhr und Echtzeituhr/ Daten konvertieren: vom C64 zum Amiga, Atari ST und PC

DTP



SH 0039: DTP, Textverarbeitung
Komplettes DTP-Paket zum Abtippen / Super Textsystem / Hochauflösendes Zeichenprogramm

64'er Magazin auf einen Blick

Diese 64'er-Ausgaben bekommen Sie noch bei Markt&Technik für jeweils 7,- DM. Die Preise für Sonderhefte und Sammelbox entnehmen Sie bitte dem Bestellcoupon. Tragen Sie Ihre Bestellung im Coupon ein und schicken Sie ihn am besten gleich los, oder rufen Sie einfach unter 089 - 20 25 15 28 an.

11/90: Bausatztest: Der Taschengeldplotter / Vergleichstest: Drucker der Spitzenklasse / 5 Schnellbauschaltungen

12/90: Abenteuer BTX / Multitasking für C64 / Großer Spieleschwerpunkt / Programmierwettbewerb: 30 000 DM zu gewinnen

1/91: Die Besten Tips&Tricks / Neu: Reparaturecke / Floppy-Flop: Betriebssystem überlistet / Jahresinhaltsverzeichnis

2/91: Sensation: Festplatte für den C 64 / Drucken ohne Ärger / Listing des Monats: Actionspiel "Ignition" / Longplay: Dragon Wars

3/91: Bauanleitung: Universelles Track-Display / Alles über Module für den C 64 / Festplatte HD 20 unter GEOS

4/91: Spiele-Schwerpunkt: 100 Tips, News, Tests / Neu: Grafikkurs / Fischer-Baukästen / Bauanleitung: Digitizer

5/91: Ätzanlage unter 50.- DM / GRB-Monitor am C64 / Longplay: Bard's Tale / Reparaturkurs: Die neuen C64 / Piratenknacker

6/91: C64er-Meßlabor: Universell erweiterungsfähig / Test: Pocket-Wrighter 3.0 - Bestes C64 Textprogramm / Listing des Monats: Autokosten im Griff

7/91: Trickfilm mit dem C 64 / Bauanleitung: 1541-Floppy mit Batteriebetrieb / Listing des Monats: Basic-Butler

8/91: Drucker unter 1000 DM / Test: GEO-RAM / Listing des Monats: 80-Farben-Malprogramm / Longplay: Secret of the Silver Plate

09/91: Joystick im Test / Die üblen Tricks mit Raubkopien / Die besten Drucker unter 1500 DM / Mit großem Spieleteil

10/91: 100 besten Tips&Tricks / Listing: Fraktal-Programm / C-64-Meßlabor: komfortables Kontrollmodul

11/91: Alles über Diskette & Floppy / Bauanleitung: C 64 steuert Laserstrahl / Sha-Jongg: Topspiel mit Spitzengrafik / Großer Spieleteil

12/91: Alle Spiele 1991 / Tolle Tips&Tricks für den C64 und C128 / Geschenktips für Computerfans / Komfortable Videoverwaltung

GRAFIK



SH 0020: Grafik
Grafik-Programmierung / Bewegungen



SH 0045: Grafik
Listings mit Pfiff / Alles über Grafik-Programmierung / Erweiterungen für Amica-Paint



SH 0055: Grafik
Amica-Paint: Malen wie ein Profi / DTP-Seiten vom C64 / Tricks&Utilities zur Hires-Grafik



SH 0063: Grafik
Text und Grafik mischen ohne Flimmern / EGA: Zeichenprogramm der Superlative / 3 professionelle Editoren



SH 0068: Anwendungen
Kreuzworträtsel selbstgemacht/ Happy Synth: Super-Synthesizer für Sound-Freaks/ Der C64 wird zum Planetarium/ Sir-Compact: Bit-Packer verdichtet Basic- und Assemblerprogramme.

SPIELE



SH 0030: Spiele für C 64 und C 128
Spiele zum Abtippen für C 64/ C 128 / Spieleprogrammierung



SH 0037: Spiele
Adventure, Action, Geschicklichkeit / Profihilfen für Spiele / Überblick, Tips zum Spielekauf



SH 0042: Spiele
Profispiele selbst gemacht / Adventure, Action, Strategie

SH 0049: Spiele
Action, Adventure, Strategie / Sprites selbst erstellen / Virenkiller gegen verseuchte Disketten



SH 0052: Abenteuerspiele
Selbstprogrammieren: Von der Idee zum fertigen Spiel / So knacken Sie Adventures

SH 0054: Spiele
15 tolle Spiele auf Diskette/ der Sieger unseres Programmierwettbewerbs: Crillion II/ ein Cracker packt aus: ewige Leben bei kommerziellen Spielen



SH 0060: Adventures
8 Reisen ins Land der Fantasie - so macht Spannung Spaß



Top Spiele 1
Die 111 besten Spiele im Test/ Tips, Tricks und Kniffe zu heißen Games/ Komplettlösung zu "Last Ninja II"/ große Marktübersicht: die aktuellen Superspiele für den C64



SH 0061: Spiele
20 heiße Super Games für Joystick-Akrobaten/ Cheat-Modi und Trainer POKEs zu über 20 Profi-Spielen/ Krieg der Kerne: Grundlagen zur Spielerprogrammierung



SH 0066: Spiele
15 Top-Spiele mit Action und Strategie/ Mondlandung: verblüffend echte Simulation und Super-Grafik/ High-Score-Knacker: Tips&Tricks zu Action-Games

BESTELLCOUPON

Ich bestelle ____ 64er Sonderhefte Nr. ______________ ____ **DM**
zum Preis von je: 14,- DM (Heft ohne Diskette) ____ **DM**
16,- DM (Heft mit Diskette) ____ **DM**
9,80 DM (SH "Top Spiele 1") ____ **DM**
24,- DM (für die Sonderhefte 0051/ 0058/ 0064) ____ **DM**

Ich bestelle ____ 64er Magazin Nr. ____ / ____ / ____ ____ **DM**
zum Preis von je 7,- DM

Ich bestelle ____ Sammelbox(en) ____ **DM**
zum Preis von je 14,- DM

Gesamtbetrag ____ **DM**

Ich bezahle den Gesamtbetrag zzgl. Versandkosten nach Erhalt der Rechnung.

Name, Vorname ______________

Straße, Hausnummer ______________

PLZ, Wohnort ______________

Telefon (Vorwahl) ______________ Ich erlaube Ihnen hiermit mir interessante Zeitschriftenangebote auch telefonisch zu unterbreiten (ggf. streichen).

Schicken Sie bitte den ausgefüllten Bestellcoupon an: 64er Leserservice, CSJ, Postfach 140 220, 8000 München 5, Telefon 089/ 20 25 15 28

Programmierkurs für Spiele

Auf dem Bildschirm erscheint eine fremde Welt: surrealistische Landschaften mit futuristischen Flugobjekten. Die Hatz geht über Schluchten, stets verfolgt und bombardiert von zahllosen Gegnern. Wer kann sich der Faszination derartiger Szenen entziehen, möchte nicht selbst so eine Fantasywelt erschaffen? Es geht.

Ein verregneter Samstagnachmittag – tote Hose. Doch zum Glück besitzen Sie ja Ihren C64. Sie legen also die Diskette mit Ihrem Lieblingsspiel ein; und nach einigen Sekunden fesselt Sie ein Spiel mit perfekter Grafik und tollem Sound.

Es ist lediglich die Kenntnis moderner Programmiertechniken nötig, um sowas selbst zu kreiern, und natürlich sollten Sie wissen, was mit Ihrem C64 machbar ist und was nicht.

Für unseren Kurs sollten Sie ein paar Kenntnisse in Assembler besitzen. Wir setzen aber kein zu hohes Niveau voraus. So genügt für noch unerfahrene Programmierer das Sonderheft 71 »Assembler« als Grundlektüre und Nachschlagewerk.

Bevor wir uns näher mit der Programmierung befassen, sehen wir uns einmal an, welche Arten von Computerspielen es gibt:

1. Adventures und Rollenspiele
... versetzt Sie in die Rolle einer anderen Person. Für diese haben Sie eine (oder mehrere) Aufgabe(n) zu lösen. Dazu wandern Sie durch Landschaften, suchen dabei in Irrgärten (Dungeons) nach versteckten Schätzen, kämpfen mit Monstern und lösen Rätsel. Eine Abwandlung sind sog. Textadventures. Bei ihnen findet ein Frage-/Antwortspiel statt.

Typische Vertreter von Adventures sind »Bard's Tale«, »Questron« oder »Detektiv 2000« in diesem Heft.

2. Geschicklichkeitsspiele
... testen und erweitern die Grenzen Ihres Reaktionsvermögens. Sie steuern ein Raumschiff mit dem Joystick durchs All oder lenken ein Fabelwesen durch finstere Gänge, wobei man Ihnen überall Hindernisse in den Weg legt, die Sie nicht mal streifen dürfen. »Mission II« in diesem Heft zeigt Ihnen deutlich was Sache ist.

3. Actionspiele
... sind meist auch zugleich Geschicklichkeitsspiele. Bei ihnen treten gehäuft Gegner auf, die Sie in einer festgesetzten Zeit vernichten müssen. Meistens lassen sie sich per Feuerknopf abschießen, und zwar in Serie. Daher nennt man diese Spieleart auch sehr ungnädig »Ballerspiele«. Die Vertreter dieser Spieleart sind primitiv bis durchaus anspruchsvoll. Beispiele für »Action« sind »Hyperthrust« oder »Flowers« in diesem Heft.

4. Jump and Run-Spiele
Bei ihnen muß der Computerspieler über Leitern oder Treppen unterschiedliche Ebenen überwinden, dabei Gegenstände durch Springen (Jump) oder Berühren aufsammeln. Das Ganze hört sich zwar einfach an, wird aber erschwert, weil bösartige Gnome die Gänge unsicher machen und jeder Kontakt mit ihnen tödlich für den Computerspieler ist. Leider sind diese Gegner nur in Ausnahmesituationen angreifbar, also hilft nichts anderes als ausreißen (and Run). Die Urversion dieser Spielephilosophie ist der berühmte »Pac Man«.

5. Strategiespiele
... läßt Sie meist durch überdimensionale Landschaften fahren oder wandern. Dabei gilt es einen oder mehrere Gegner zu vernichten. Im Gegensatz zum Adventure aber nicht durch Rätsel, sondern durch Steuerung des Kampfgeschehens. Das kann wie bei »Tron« in diesem Heft unkriegerisch, mit dem Joystick geschehen, oder wie bei »Field of Fire« in mehreren Kampfphasen. In der ersten Phase, der Bewegungsphase, bewegt der Spieler seine Truppen und der Computer die gegnerischen. Anschließend folgt die Feuerrunde, in der die zu vernichtenden Gegner ausgesucht werden. Danach werden die Ergebnisse dieser Aktionen berechnet und angezeigt.

6. Brettspiele
... simulieren herkömmlich mechanische Spiele auf dem Computer. Die Palette reicht dabei von »Monopoly« bis »Schach«.

7. Simulationen
... versuchen einen Ausschnitt der Wirklichkeit auf dem Computer darzustellen. Man unterscheidet dabei zwischen zwei grundlegenden Veriationen:

1. Wirtschafts-Simulationen, wie z.B. »Omnibus« in diesem Heft. Sie entscheiden über Gedeih oder Verderb einer Firma.
2. Technische Simulationen. Bei diesen verändern Sie Konstruktionsmerkmale und beeinflussen so die Funktion des simulierten Geräts.

Wie sieht ein Spiel aus

Es gibt also Mischformen und meist lassen sich einzelne Spiele auch nicht eindeutig zuordnen. Wie auch immer – neue Ideen finden Anklang (wenn sie gut realisiert sind).Für unseren Kurs haben wir Geschicklichkeits- und Actionspiele ausgewählt, denn diese sind am eindrucksvollsten anzusehen und vereinigen die meisten Programmiermerkmale aller anderen Spielarten in sich. Überlegen wir uns, was wir für eine typische Spieleszene benötigen:

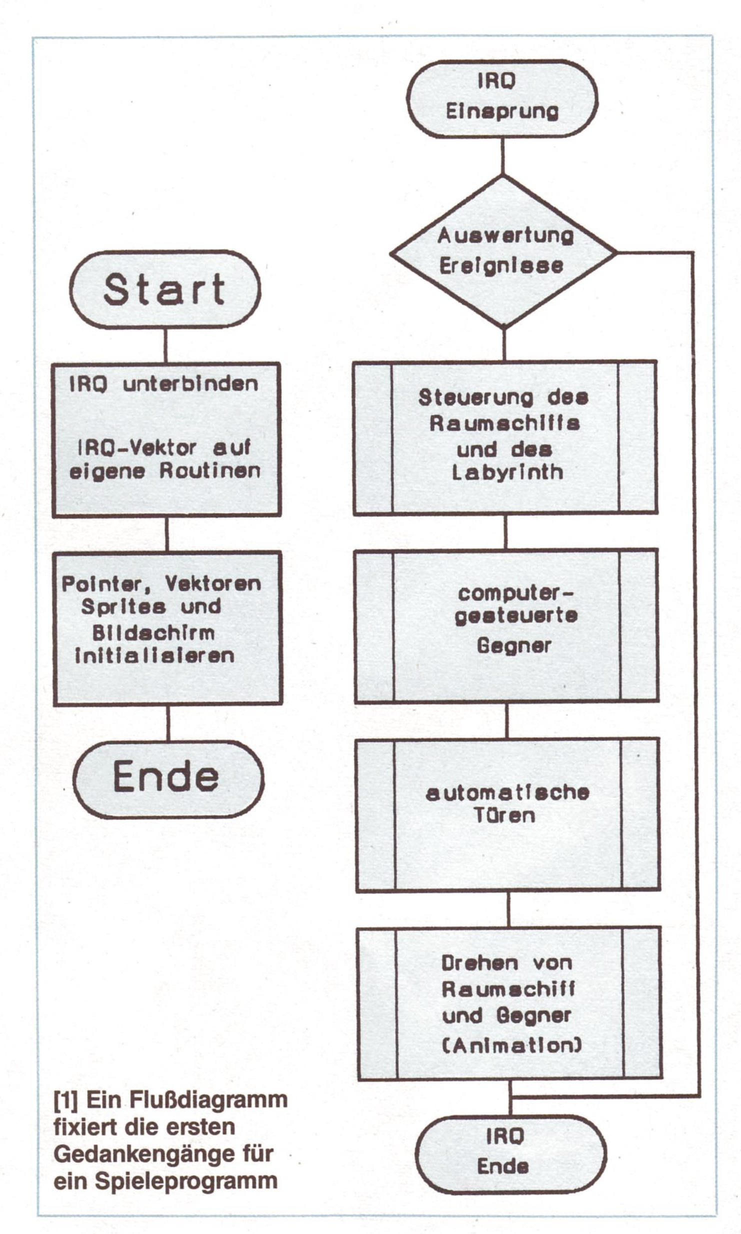
Grafik ... das Design der Spiele-Landschaft
handelnde Figuren ... auch ihr Design muß festgelegt werden
Geräusche
... bei einer außergewöhnlichen Aktion erhöht ein besonderes Geräusch die Spannung
Hintergrundmusik
... dient zur Unterhaltung und gehört aus dramaturgischen Gründen zu jedem Spiel.
Spielablauf
... ist das schwierigste am ganzen Spiel. Er übersetzt Ihre Idee in die Spielbarkeit. Die Grafik kann noch so formvollendet, die Musik noch so perfekt sein, erst der Ablauf entscheidet über die Brauchbarkeit des Spiels.

Packen wir's an:
Wir werden ein Fantasieraumschiff durch ein Labyrinth steuern. Da dies zwar schön, aber zu einfach ist, lassen wir Hindernisse auftreten. Computergesteuerte Gegner und wie in einem Labyrinth üblich, noch etliche Hindernisse. Interessant sind auch Türen, die sich zyklisch öffnen und schließen und bei dieser Gelegenheit ist es vielleicht noch interessant, das Raumschiff ständig zu drehen.

Soweit zur Grundidee. Wir wollen sie in der ersten Phase schon mal festhalten – am besten in einem Ablaufplan (Abb.1, S. 16).

Danach ist es angebracht, einige Nächte drüber zu schlafen um sicherzugehen, daß alles Gewünschte mit eingeplant ist. Natürlich lassen sich kleine Verbesserungen auch noch am fertigen oder fast fertigen Spiel vornehmen, aber grund-


DIE
SPIELHÖLLE
KURS
NEBEN DER
KÜCHE



[1] Ein Flußdiagramm fixiert die ersten Gedankengänge für ein Spieleprogramm

sätzliche Änderungungen sind nachträglich sehr schwer einzubauen.

Da der Ablaufplan nur eine grobe Hilfe ist, sollte der nächste Arbeitsschritt das Vorbereiten fertiger Bestandteile des Spiels sein. Dazu sollten Sie eine Diskette formatieren, auf die dann alle fertig erarbeiteten Teile gespeichert werden. Für unseren Kurs haben wir Ihnen alle Bestandteile schon in einem Programm zusammengefügt. Laden Sie es mit

```
LOAD"DEMO*",8
```

und starten Sie mit RUN. Das File wird anschließend entpackt und einzelne Programmteile, wie Grafik, Sprites, usw. werden an die richtigen Speicherstellen geschoben. Starten Sie aber noch nicht (SYS12288).

Damit Sie sowohl Zeichensatz als auch Sprites und Spiele-Landschaft umgestalten und in eigenen Spielen verwenden können, haben wir Ihnen einen Maschinensprache-Monitor mit auf Diskette gespeichert. Sie laden ihn mit

```
LOAD"SMON $C000",8,1
```

Danach geben Sie NEW ein. Gestartet wird der SMON mit SYS49152. Die Befehlsbeschreibung erfahren Sie aus dem Textkasten »Quickreferenz SMON« auf S. 24. Für unsere Arbeits-Diskette lassen sich die einzelnen Bestandteile des Spiels abspeichern (bis auf die Musik, sie liegt genau an der Position, wo sich jetzt der SMON befindet):

Für die Sprites geben Sie ein:

```
S"SPRITES"0800 0C00
```

Die Grafik wird mit

```
S"LANDSCHAFT"7000 8A00
```

auf Diskette gebracht. Schließlich benötigen Sie noch den geänderten Zeichensatz:

```
S"ZEICHENSATZ"2000 3000
```

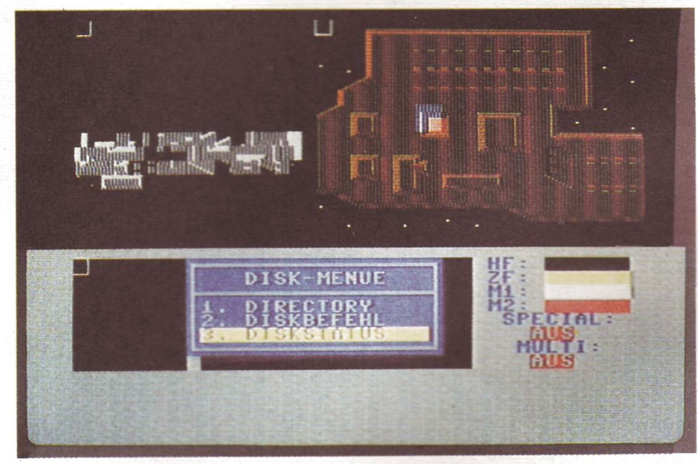
verewigt auch ihn auf der Arbeitsdiskette.

Die Grafik

Zunächst muß das Labyrinth aufgebaut werden. Das könnte man mit einer hochauflösenden Grafik realisieren, z.B. über den Umweg einer Zeichnung mit Hi-Eddi oder Koala-Painter. Diese Möglichkeit ist sicherlich nicht ideal, da eine Bitmap 8 KByte (Kilobyte, 8 x 1024 Byte) Speicher kostet. Für Spiele gibt es daher eine speicherplatzgünstigere Alternative: den normalen Zeichensatz-Modus des VIC (Videochip). Dieser benötigt nur 1 KByte für das Video-RAM.

Wenn Sie Spiele mit dem Standardzeichensatz des C64 aufbauen, bleiben sie leider nur relativ eintönig. Bedeutend mehr Möglichkeiten bieten eigene Zeichensätze. Sie kennen sicherlich die Fähigkeit des C64, umdefinierte Zeichensätze zu verwenden. Normalerweise holt sich der Video-Interface-Chip (VIC) seine Informationen aus einem Nur-Lese-Speicher mit unveränderlichem Dateninhalt (ROM). In diesem sind die im Handbuch beschriebenen Bildschirmcode-Zeichenmuster unverrückbar gespeichert. Aber der VIC ist ein sehr flexibler Baustein. Er kann seine Buchstabenmuster auch aus anderen Bereichen des Speichers holen. Man muß ihm nur sagen, wo die Daten liegen.

Ein Zeichensatz benötigt 2 KByte. Er alleine nützt uns allerdings nichts. Die Zusammenstellung der einzelnen Zeichen auf dem Bildschirm erst ergibt ein exaktes Bild des »Screens« (Bildschirms). Dafür ist das Video-RAM zuständig. Es liegt normalerweise ab Speicherstelle 1024 ($0400) im Speicher und enthält 1000 Zeichen. Jeweils 40 nebeneinander ergeben eine Bildschirmzeile und 25 Zeilen ergeben einen kompletten Bildschirm. Video-RAM und Zeichensatz benötigen



[2] Zeichensätze lassen sich komfortabel mit dem »Character-Editor« aus Sonderheft 55 gestalten

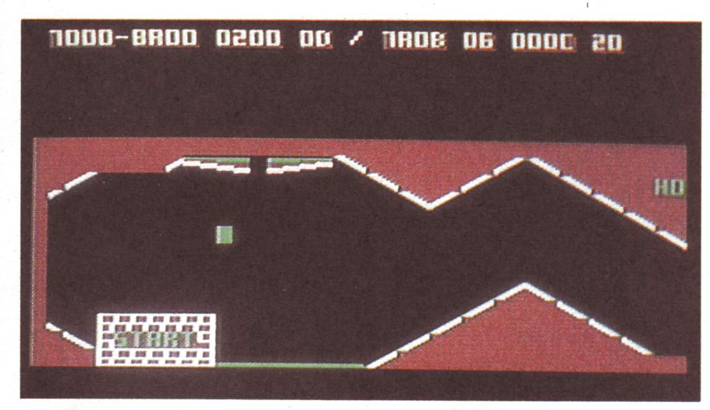
zusammen 3 KByte. Wie Sie sehen, wesentlich weniger als eine Bitmap.

Schaffen wir also zuerst die Voraussetzung für unsere Höhlenlandschaft – den sog. Spielezeichensatz. Kreiert wird er mit einem Zeichen-Editor (z.B. dem »Character-Editor« aus Sonderheft 55, Abb.2). Dieser Editor besitzt neben den Feldern zur Buchstabenumgestaltung einen Bereich, auf dem die Kombinationen der einzelnen Zeichen ausprobiert werden können. Dadurch läßt sich recht komfortabel ein neuer

Zeichensatz konstruieren und ausprobieren. Sie sollten ihn auf die Arbeitsdiskette speichern.

Wie konstruieren wir unsere Grafiklandschaft? Wir haben ja beschlossen, durch eine Höhlenlandschaft zu reisen. Dazu ist eine Bildschirmgröße alleine aber zu wenig; zu schnell würde man am Rand anstoßen. Wir benötigen also ein wesentlich längeres Labyrinth, als es der C64 auf einem Bildschirm darstellen kann. Später verschieben wir den sichtbaren Bereich kontinuierlich und erwecken so den Eindruck, als würden wir uns durch die Landschaft bewegen. Der Fachbegriff für diese Art der Hintergrundbewegung ist »scrolling«. Und noch etwas sollten wir schon bei der Konstruktion berücksichtigen:

Am oberen und unteren Bildschirmrand müssen ein paar Zeilen frei bleiben, für die Punkteanzeigen (score) usw. Für uns bedeutet dies: Wir konstruieren eine Landschaft, die mit dem eigentlichen Bildschirmformat nichts mehr zu tun hat. Achten Sie ebenfalls schon bei der Konstruktion darauf, daß bei den Zeichen, über die später unser Raumschiff fliegen soll, keine Pixel gesetzt sein dürfen. Wir überprüfen später im Programm, ob unsere Spielefigur mit dem Hintergrund zusammengestoßen ist. Daher dürfen die Gänge, durch die wir fahren wollen, auch keine Hintergrundinformationen enthalten. Im normalen Commodore-Zeichensatz ist dies das Leerzeichen mit dem Bildschirmcode »32« ($20). Um Ihnen die Konstruktion von Riesenbildschirmen zu erleichtern, befindet sich ein Editor mit auf der beiliegenden Diskette. Die Bedienungsanleitung finden Sie im Textkasten »der Grafikeditor«.



[3] Mit dem »Grafik Editor« ist die Gestaltung des Spielfeldes ein Kinderspiel. Sowohl Speicheranfang, als auch Größe lassen sich dem eigenen Spielfeld anpassen.

Die Spielfiguren

Als nächstes werden Helden und Fieslinge benötigt, also unser Raumschiff und die Gegner. Hierfür nutzen wir die Fähigkeit des C64, sog. Sprites darzustellen. Sprites sind eigene kleine Grafiken, die unabhängig vom anderen Bildschirminhalt dargestellt werden. Ihr Format ist auf 24 x 21 Punkte (einfarbig) oder 12 x 21 Punkte (mehrfarbig) festgelegt und es können (ohne Tricks) maximal acht davon zugleich erscheinen. Man bastelt sich diese Sprites mit einem Sprite-Editor (Sonderheft 33), oder klaut sie sich aus anderen Programmen und verändert sie nach eigenen Wünschen (»Sprite-Control«, Sonderheft 55).

Eines sollten wir noch berücksichtigen: Wir wollen unser Raumschiff ständig drehen. Dazu lassen wir später (wie im Film) mehrere Bilder nacheinander ablaufen (Animation). Geschieht dies in der richtigen Geschwindigkeit, wird dem Auge der Eindruck einer fließenden Bewegung vermittelt. Beim Konstruieren der Sprites müssen wir dies allerdings schon überlegen: Die Anzahl der nötigen Bilder muß festgelegt werden, und natürlich müssen die Sprites gezeichnet werden.

So finden Sie die Programme auf der Diskette

DISKETTE SEITE 1

```
0   "64'ER SONDERH.73" S1
0      "┌──────────────┐" DEL
0      "| GESCHICK-    |" DEL
0      "|  LICHKEIT    |" DEL
0      "└──────────────┘" DEL
39     "MISSION II"       PRG  Seite 4
38     "BY BLUE CHIP"     PRG
17     "MISS00"           PRG
12     "MISS01"           PRG
20     "MISS02"           PRG
21     "MISS03"           PRG
28     "MISS04"           PRG
15     "MISS05"           PRG
11     "MISS06"           PRG
13     "MISS07"           PRG
10     "MISS08"           PRG
11     "MISS09"           PRG
5      "MISS10"           PRG
14     "TRAINER.CDE"      PRG
0      "┌──────────────┐" DEL
0      "|    ACTION    |" DEL
0      "└──────────────┘" DEL
3      "HYPERTHRUST"      PRG  Seite 39
1      "HYPERSCORE"       PRG
0      "┌──────────────┐" DEL
0      "|  STRATEGIE   |" DEL
0      "└──────────────┘" DEL
27     "TRON"             PRG  Seite 45
0      "┌──────────────┐" DEL
0      "|     KURS     |" DEL
0      "└──────────────┘" DEL
27     "DEMOGAME"         PRG  Seite 14
0      "----------------" DEL
4      "SOUND"            PRG
8      "LIED"             PRG
0      "----------------" DEL
17     "SMON $C000"       PRG
0      "┌──────────────┐" DEL
0      "|   DISKETTE   |" DEL
0      "|  BEIDSEITIG  |" DEL
0      "|   BESPIELT   |" DEL
0      "└──────────────┘" DEL
```

DISKETTE SEITE 2

```
0   "64'ER SONDERH.73" S2 2A
0      "┌──────────────┐" DEL
0      "|     KURS     |" DEL
0      "└──────────────┘" DEL
12     "EDITOR"           PRG  Seite 14
0      "┌──────────────┐" DEL
0      "|    ACTION    |" DEL
0      "└──────────────┘" DEL
7      "FLOWERS"          PRG  Seite 38
0      "┌──────────────┐" DEL
0      "|    SPORT     |" DEL
0      "└──────────────┘" DEL
27     "W.P TENNIS II"    PRG  Seite 37
0      "┌──────────────┐" DEL
0      "| GESCHICK-    |" DEL
0      "|  LICHKEIT    |" DEL
0      "└──────────────┘" DEL
40     "ROLL IT!"         PRG  Seite 5
0      "----------------" DEL
64     "MIC'S PUSH 'EM"   PRG  Seite 44
0      "----------------" DEL
62     "ERUCA"            PRG  Seite 6
0      "┌──────────────┐" DEL
0      "|   PREVIEW    |" DEL
0      "└──────────────┘" DEL
67     "TILT /TC"         PRG  Seite 7
0      "┌──────────────┐" DEL
0      "| WIRTSCHAFTS- |" DEL
0      "| SIMULATION   |" DEL
0      "└──────────────┘" DEL
17     "OMNIBUS GMBH"     PRG  Seite 40
109    "OMNI.MAIN"        PRG
40     "OMNI.PIC"         PRG
9      "OMNI.CHAR"        PRG
1      "OMNI.END"         SEQ
3      "OMNI.ABSCHLUSS"   PRG
10     "OB.VIER"          SEQ
0      "┌──────────────┐" DEL
0      "|  ADVENTURE   |" DEL
0      "└──────────────┘" DEL
154    "DETEKTIV 2000"    PRG  Seite 42
0      "┌──────────────┐" DEL
0      "| BASIC        |" DEL
0      "| ERWEITERUNG  |" DEL
0      "└──────────────┘" DEL
9      "GAME-BASIC"       PRG  Seite 46
13     "GAMEBASIC-GAME"   PRG
10     "GAMEBASIC-DEMO"   PRG
9      "ZSA"              PRG
0      "┌──────────────┐" DEL
0      "|     ENDE     |" DEL
0      "└──────────────┘" DEL
```

WICHTIGE HINWEISE

zur beiliegenden Diskette:

Aus den Erfahrungen der bisherigen Sonderhefte mit Diskette wollen wir ein paar Tips an Sie weitergeben:

1 **Bevor Sie mit den Programmen auf der Diskette arbeiten, sollten Sie unbedingt eine Sicherheitskopie der Diskette anlegen. Verwenden Sie dazu ein beliebiges Kopierprogramm, das eine komplette Diskettenseite dupliziert.**

2 **Auf der Originaldiskette ist wegen der umfangreichen Programme nur wenig Speicherplatz frei. Dies führt bei den Anwendungen, die Daten auf die Diskette speichern, zu Speicherplatz-Problemen. Kopieren Sie daher das Programm, mit dem Sie arbeiten wollen, mit einem File-Copy-Programm auf eine leere, formatierte Diskette und nutzen Sie diese als Arbeitsdiskette.**

3 **Die Rückseite der Originaldiskette ist schreibgeschützt. Wenn Sie auf dieser Seite speichern wollen, müssen Sie vorher mit einem Diskettenlocher eine Kerbe an der linken oberen Seite der Diskette anbringen, um den Schreibschutz zu entfernen. Probleme lassen sich von vornherein vermeiden, wenn Sie die Hinweise unter Punkt 2 beachten.**

ALLE PROGRAMME
aus diesem Heft



HIER

Herausgeber: Carl-Franz von Quadt, Otmar Weber

Redaktionsdirektor: Dr. Manfred Gindle

Chefredakteur: Georg Klinge – verantwortlich für den redaktionellen Teil
Stellv. Chefredakteur: Arnd Wängler
Textchef: Jens Maasberg
Produktion: Andrea Pfliegensdörfer
Redaktion: Harald Beiler (bl), Herbert Großer (gr)
Redaktionsassistenz: Sylvia Wilhelm, Birgit Misera (089/4613202)
Telefax: 089/4613-5001
Alle Artikel sind mit dem Kurzzeichen des Redakteurs und/oder mit dem Namen des Autors/Mitarbeiters gekennzeichnet

Manuskripteinsendungen: Manuskripte und Programmlistings werden gerne von der Redaktion angenommen. Sie müssen frei sein von Rechten Dritter. Sollten sie auch an anderer Stelle zur Veröffentlichung oder gewerblichen Nutzung angeboten worden sein, muß dies angegeben werden. Mit der Einsendung von Manuskripten und Listings gibt der Verfasser die Zustimmung zum Abdruck in von der Markt&Technik Verlag AG herausgegebenen Publikationen und zur Vervielfältigung der Programmlistings auf Datenträger. Mit der Einsendung von Bauanleitungen gibt der Einsender die Zustimmung zum Abdruck in von Markt&Technik Verlag AG verlegten Publikationen. Honorare nach Vereinbarung. Für unverlangt eingesandte Manuskripte und Listings wird keine Haftung übernommen.

Verlagsleitung: Wolfram Höfler
Operation Manager: Michael Koeppe

Layout: Dagmar Portugall
Bildredaktion: Walter Linne (Fotografie); Ewald Standke, Norbert Raab (Spritzgrafik); Werner Nienstedt (Computergrafik)

Anzeigendirektion: Jens Berendsen
Anzeigenleitung: Philipp Schiede (399) – verantwortlich für die Anzeigen
Telefax: 089/4613-775

Anzeigenverwaltung und Disposition: Chris Mark (421)

Auslandsrepräsentation:
Auslandsniederlassungen:
Schweiz: Markt&Technik Vertriebs AG, Kollerstr. 37, CH-6300 Zug, Tel. 0041/42-440550, Telefax 0041/42-415770
USA: M&T Publishing Inc.; 501 Galveston Drive Redwood City, CA 94063, Telefon: (415) 366-3600, Telefax 415-3663923
Österreich: Markt&Technik Ges. mbH, Große Neugasse 28, A 1040-Wien Telefon: 0043/1/5871393O, Telefax: 0043-1-5871 39333
Anzeigen-Auslandsvertretung:
Großbritannien: Smyth Int. Media Representatives, Telefon 0044/81340-5058, Telefax 0044/81341-9602

Israel: Baruch Schaefer, Telefon 00972-3-5562256, Telefax 00972/52/444518
Taiwan: AIM Int. Inc., Telefon 00886-2-7548613, Telefax 00886-2-7548710

Japan: Media Sales Japan, Telefon 0081/33504-1925, Telefax 0081/33596-1709
Korea: Young Media Inc., Telefon 0082-2-7564819, Telefax 0082-2-7575789
Frankreich: CEP France, Telefon 0033/148007616, Telefax 0033/148240202
Italien: CEP Italia, Telefon 0039/24982997, Telefax 0039/24692834
International Business Manager: Stefan Grajer 089/4613-638

Gesamtvertriebsleiter: York v. Heimburg
Vertriebsmarketing: Helmut Pleyer (710)

Vertrieb Handel: Inland (Groß-, Einzel- und Bahnhofsbuchhandel) sowie Österreich und Schweiz: ip Internationale Presse, Ludwigstraße 26, 7000 Stuttgart 1, Tel. 0711/619660

Einzelheft-Bestellung: Markt & Technik Leserservice, CSJ Postfach 140220, 8000 München 5

Verkaufspreis: Das Einzelheft kostet DM 16,–

Produktion: Klaus Buck (Ltg./180), Wolfgang Meyer (Stellv./887);

Druck: SOV Graphische Betriebe, Laubanger 23, 8600 Bamberg

Sonderdruck-Dienst: Alle in dieser Ausgabe erschienenen Beiträge sind in Form von Sonderdrucken zu erhalten. Anfragen an Reinhard Jarczok, Tel. 089/4613-185, Fax 4613-774.

Vorstand: Otmar Weber (Vors.), Bernd Balzer, Dr. Rainer Doll, Lutz Glandt

Direktor Zeitschriften: Michael M. Pauly

Anschrift für Verlag, Redaktion, Vertrieb, Anzeigenverwaltung und alle Verantwortlichen: Markt&Technik Verlag Aktiengesellschaft, Hans-Pinsel-Straße 2, 8013 Haar bei München, Telefon 089/4613-0, Telex 522052, Telefax 089/4613-100

ISSN 0931-8933

Telefon-Durchwahl im Verlag:
Wählen Sie direkt: Per Durchwahl erreichen Sie alle Abteilungen direkt. Sie wählen 089/4613 und dann die Nummer, die in den Klammern hinter dem jeweiligen Namen angegeben ist.

Mitteilung gem. Bayerischem Pressegesetz:
Aktionäre, die mehr als 25% des Kapitals halten: Otmar Weber, Ingenieur, München; Carl-Franz von Quadt, Betriebswirt, München; Aufsichtsrat: Carl-Franz von Quadt (Vorsitzender), Dr. Robert Dismann (stellv. Vorsitzender), Dr. Erich Schmitt

Copyright-Erklärung

Name: ....
....

Anschrift: ....

Datum: ....

Computertyp: ....

Benötigte Erweiterung/Peripherie: ....

Datenträger: Kassette/Diskette ....

Programmart: ....

Ich habe das 18. Lebensjahr bereits vollendet

...., den ....

....
(Unterschrift)

Wir geben diese Erklärung für unser minderjähriges Kind als dessen gesetzliche Vertreter ab.

...., den ....

....

Bankverbindung:

Bank/Postgiroamt: ....

Bankleitzahl: ....

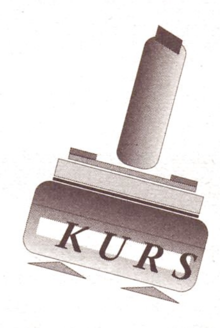
Konto-Nummer: ....

Inhaber des Kontos: ....

Das Programm/die Bauanleitung: ....

das/die ich der Redaktion der Zeitschrift 64'er übersandt habe, habe ich selbst erarbeitet und nicht, auch nicht teilweise, anderen Veröffentlichungen entnommen. Das Programm/die Bauanleitung ist daher frei von Rechten anderer und liegt zur Zeit keinem anderen Verlag zur Veröffentlichung vor. Ich bin damit einverstanden, daß die Markt & Technik Verlag AG das Programm/die Bauanleitung in ihren Zeitschriften oder ihren herausgegebenen Büchern abdruckt und das Programm/die Bauanleitung vervielfältigt, wie beispielsweise durch Herstellung von Disketten, auf denen das Programm gespeichert ist, oder daß sie Geräte und Bauelemente nach der Bauanleitung herstellen läßt und vertreibt bzw. durch Dritte vertreiben läßt.

Ich erhalte, wenn die Markt & Technik Verlag AG das Programm/die Bauanleitung druckt oder sonst verwertet, ein Pauschalhonorar.



Zero-Page Adressen

$91/$92 - 145/146 - Bandoperationen
$9E bis $A2 - 158 bis 162 - Bandoperationen
$A7 bis $AE - 167 bis 171 - Band Ein-/Ausgabe
$FC bis $FF - 252 bis 255 - Vorsicht: findet meist bei Musikroutinen Verwendung

Der Grafikeditor

Er dient zum Erzeugen einer Grafiklandschaft für Scrollhintergründe und zeigt die Landschaft so, wie sie später im Spiel am Bildschirm sichtbar ist (Abb. 3). In unserem Kurs liegt der Scrollhintergrund im Speicher ab $7000 und hat eine Größe von 512 x 13 Zeichen. Dadurch ergibt sich eine Länge bis Speicherposition $8A00. Die Grafik ist linear abgelegt, d.h. die oberste Reihe der Grafik (512 Zeichen lang) ist ab $7000 bis $71FF abgelegt, die zweite Reihe von $7200 bis $73FF und so weiter. Für die Verwendung in eigenen Programmen wurden einige Zusatzfunktionen wie Ändern von Länge und Höhe der Grafik und des Grafikbeginns eingebaut. Beachten Sie, daß bei einer Formatänderung keine Umrechnung auf das neue Format stattfindet. Wenn Sie bei einer fertigen Grafik die Länge oder Höhe ändern, kommt daher auch das Aussehen der Landschaft durcheinander.

Zum Einüben der Funktionen gehen Sie folgendermaßen vor: Laden Sie zuerst das komprimierte Spiel von der Seite 1 der beiliegenden Diskette

LOAD"DEMOGAME",8

und entpacken Sie es mit RUN. Danach laden Sie den Editor mit

LOAD"EDITOR",8,1

und geben »NEW« ein. Der Editor wird mit »SYS49152« gestartet.

Er beherrscht zwei Betriebsarten:

1. Editiermodus

... ist unmittelbar nach dem Starten eingestellt. Jedes eingegebene Zeichen erscheint an der Cursorposition. Der Cursor wird wie gewöhnlich mit den Cursortasten bewegt und bei Überschreitung des rechten oder linken Bildschirmrands scrollt der Inhalt entsprechend. Die erste Zeile ist eine Statuszeile und enthält von links nach rechts folgende Informationen in hexadezimaler Schreibweise:

7000-8A00 - Speicherbeginn und Speicherende der bearbeiteten Grafik.

0200 - Länge einer Grafikzeile

0D - Höhe der Grafik. Sie darf $01 bis $18 (= max. 24 Zeilen) betragen.

7000 - aktuelle Cursorposition

01 - aktuelle Cursorzeile

0001 - aktuelle Cursorspalte

62 - Zeichencode unter dem Cursor

Die letzten vier Werte verändern sich beim Bewegen des Cursors. Die Werte Speicherbeginn, Länge und Höhe können später im Befehlsmodus geändert werden. Das Speicherende wird dabei automatisch umgerechnet. Es findet keine Plausibilitätsüberprüfung statt: Das bedeutet, daß Sie eine Superlandschaft einstellen können, die alle Speichergrenzen überschreitet. Da dies natürlich nicht sinnvoll ist, sollten Sie bei Parameteränderungen auf das Speicherende achten. Ebenso können die RAM-Bereiche unter dem ROM zwar beschrieben, aber nicht sichtbar gemacht werden.

Bei Überschreitung der erlaubten Grafikhöhe führt das Programm ein RESTORE durch und muß mit SYS49152 neu gestartet werden. Achtung: Weder Grafikhöhe noch -Länge dürfen auf Null gesetzt sein.

Im Editor ist Groß- und Kleinschrift über die Tastatur erreichbar, alle anderen Zeichen über eine spezielle Option:

<CONTROL> und Zahl

... bietet die Möglichkeit über den Bildschirmcode ein Zeichen auf die Cursorposition zu setzen. Dazu drücken Sie <CONTROL> und halten diese Taste gedrückt. Bei gedrückter Taste tippen Sie über die Zifferntasten (<0> bis <9> und <A> bis <F>) zweistellig den gewünschten Bildschirmcode (hexadezimal) ein. Nach dem Loslassen von <CONTROL> erscheint das gewünschte Zeichen an der Cursorposition.

<CLR/HOME>

... bringt Sie immer an den Speicheranfang des Scrollhintergrunds

<F8>

... bietet eine Repeat-Funktion des letzten getippten Zeichens an der aktuellen Cursorposition (bei Programmstart mit $20). Da die Tastaturwiederholung eingeschaltet ist, lassen sich so auf einfachste Weise Linien oder Flächen erzeugen.

<SHIFT CLR/HOME>

... füllt die gesamte Bildschirmlandschaft mit dem letzten getippten Zeichen (bei Programmstart mit $20).

<F4>

... schaltet um zwischen Multicolor- oder Single-Farb-Modus.

<F6>

... schaltet um zwischen Commodore oder geändertem Zeichensatz. Der geänderte Zeichensatz muß ab $2000 (8192) beginnen. Eine Ladefunktion für Zeichensätze ist nicht vorhanden, Sie müssen den Zeichensatz also mit einem Maschinensprache-Monitor oder einem Basic-Lader an diese Speicherposition laden.

<F1>

... ändert die Zeichenfarbe. Jeder Tastendruck erhöht die Farbnummer (s. Handbuch). Beachten Sie: Im Multicolor-Modus dürfen für die Zeichenfarbe keine Farbnummern <8 verwendet werden, sonst schaltet das Video-Interface-Chip zurück auf Single-Farb-Modus.

<F3>

... ändert die Hintergrundfarbe (Register 53281).

<F5>

... ändert die Multicolorfarbe 1 (Register 53282).

<F7>

... ändert die Multicolorfarbe 2 (Register 53283).

<F2>

... verläßt das Programm nach Basic. Einen Neustart erreichen Sie mit SYS49152, dabei bleiben Speicherinhalt und Parameter-Einstellungen erhalten (lediglich VIC-Modus und Registerfarben sind nicht gespeichert).

<RUN/STOP>

... schalten um auf Befehlsmodus

2. Befehlsmodus

In diesem Modus lassen sich Funktionen wie Laden, Speichern, Directory und Parameteränderungen vornehmen.

<F1> - Laden

... lädt eine Grafiklandschaft nach Namenseingabe von Diskette.

<F2> - Speichern

... speichert eine Grafiklandschaft nach Namenseingabe auf Diskette. Die Parameter Speicherbeginn und Speicherende werden dabei als Anfang und Endadresse angenommen.

<F3> - Directory

... gibt das Inhaltsverzeichnis einer eingelegten Diskette am Bildschirm aus.

<F4> - Länge der Grafik

... legt die Länge (horizontal) der Grafiklandschaft in Zeichen fest. Voreingestellt sind 512 Zeichen ($0200).

<F6> - Höhe der Grafik

... legt die Höhe (vertikal) der Grafiklandschaft in Zeilen. Jede dieser Zeilen hat eine Länge, wie unter <F4> festgelegt. Voreingestellt sind »0D« Zeichen (dezimal 13). Erlaubt sind »01« bis »12« (dezimal 24) Zeilen.

<F8> - Speicherbeginn

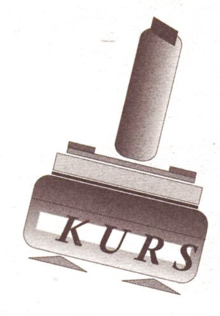
... bestimmt den Anfang der Grafiklandschaft. Ab dieser Speicherposition wird die Grafik abgelegt. Voreingestellt ist $7000. Beachten Sie: $C000 bis $CFFF sind zwar möglich, aber nicht erlaubt, da sich hier das Editierprogramm selbst befindet. Der Editor liest keine Zeichen unter den ROMs. Falls Sie in Ihrem Programm diese Bereiche verwenden wollen, bietet es sich auch an, die Speicherlandschaft in einem anderen Speicherbereich zu kreieren, auf Diskette zu speichern und beim Spiel an die richtige Speicheradresse laden.

<RUN/STOP>

... schaltet zurück zum Editiermodus.

Kurzinfo: Editor

Programmart: Editor für Grafiklandschaften
Laden: LOAD"EDITOR",8,1
Starten: nach dem Laden NEW und SYS49152 eingeben
Benötigte Blocks: 12 Blocks
Programmautor: Herbert Großer



Die Soundroutine

Erst richtig eingesetzte Musik und Geräusche geben einem Spiel den richtigen Touch und überbrücken langweilige Passagen – alter Regietrick, siehe Autoverfolgungsjagden. Damit Sie das Rad nicht zweimal erfinden müssen, haben wir Ihnen dafür eine kurze Routine auf Diskette mitgeliefert. Geladen wird sie mit

LOAD"SOUND",8,1

Danach rückt »NEW« die Basic-Pointer wieder gerade. Initialisiert wird mit »SYS491252«. Die Routine bindet sich dabei automatisch in den Interrupt ein und Musik läuft im Hintergrund ab.

Doch für Spiele ist nicht nur Musik wichtig, sondern auch Geräusche. Beispielsweise sollte für ein schießendes Raumschiff auch ein entsprechendes Geräusch aus dem Lautsprecher tönen. Für unsere Musikroutine lassen sich bis zu vier Geräusche programmieren. Zwei sind in der Routine »Sound« schon vorbereitet. Geben Sie doch mal bei laufender Musik ein:

POKE49165,1

Ein Schuß dröhnt aus dem Lautsprecher und anschließend spielt die Musik weiter. Mit »POKE49165,2« erreichen Sie in unserem Demo eine Tonfolge (»Tirili«). Die anderen Effekte sind momentan nicht belegt.

Natürlich nützt die schönste Musik- und Soundroutine nichts, wenn man nicht auch eigene Musikstücke verwenden kann. Bei unserer Soundroutine ist das möglich. Zudem läßt sich in Maschinensprache mit Overlay-Technik (nachladen anderer Files) arbeiten, d.h. für andere Spiele-Level sind auch andere Musik- und Soundeffekte nachladbar. Als Programmierbeispiel befindet sich ein File mit auf der beiliegenden Diskette. Bevor Sie es laden, sollten Sie die Musikroutinen mit »SYS49152« ausschalten. Danach lädt

LOAD"MUSIK",8

das File. Sie starten mit RUN. Lassen Sie sich nicht durch den »OUT OF DATA ERROR ?« irritieren. Da die Musik beliebig erweitert oder geändert werden kann, wurde die Anzahl der Datazeilen nicht festgelegt. Bei der Abarbeitung gehen dem Programm also irgendwann die Data aus, und das wird natürlich mit einer Fehlermeldung quittiert. Keine Angst, das Musikstück ist trotzdem komplett im Speicher. Bevor Sie ans Komponieren gehen, sollten Sie sich das Listing einmal genauer ansehen (LIST):

Zeile 230 bis 260 enthalten die Daten von vier Sound-Datensätzen. Sie lassen sich beliebig jedem Ton zuordnen. Das Format entspricht dem einer Stimme:

1. Byte – Tastverhältnis, Low-Byte

... regelt zusammen mit dem vierten Byte das Tastverhältnis bei der Wellenform Rechteck. Bei anderen Wellenformen ist der Wert beliebig.

2. Byte – Tastverhältnis, High-Byte

... regelt zusammen mit dem dritten Byte das Tastverhältnis bei der Wellenform Rechteck. Auch dieser Wert ist bei anderen Wellenformen beliebig.

3. Byte – Wellenform

... ein gesetztes Bit hat folgende Funktion:

Bit 0 (1) = Ton Ein, muß Null sein.
Bit 1 (2) = Synchronisation
Bit 2 (4) = Ringmodulation
Bit 3 (5) = Test, muß Null sein
Bit 4 (16) = Dreieck
Bit 5 (32) = Sägezahn
Bit 5 (64) = Rechteck (s. 1.Byte und 2.Byte)
Bit 5 (128) = Rauschen

Die Wellenformen lassen untereinander mischen, lediglich »Rauschen« darf nicht mit anderen Formen zusammen erscheinen. Bit 0 (Ton Ein) wird von der Abspielroutine behandelt und darf nicht gesetzt sein, da sich der Ton sonst nicht mehr über die Sounddaten ausschalten läßt. Da Bit 3 (Testbit) einen Reset der Stimme durchführt und alle anderen Bits dabei ignoriert, darf es nicht gesetzt sein. Auch dieses Bit wird von der Routine behandelt.

4. Byte – Attack/Decay

... setzt die Anschlag-, bzw. Abschwellzeit des Tons. Attack (Anschlagzeit) benötigt dafür die unteren vier Bit (0 bis 15), Decay (Abschwellzeit) die oberen (0 bis 15 x 16).

5. Byte – Sustain/Release

... bestimmt Haltelautstärke bzw. Ausklingzeit des Tons. Sustain (Haltelautstärke) benötigt dafür die oberen vier Bits (0 bis 15 x 16), Release (Ausklingzeit) die unteren vier Bit (0 bis 15).

Achtung: Sounddaten müssen jeweils mit vier Datensätzen zu je 5Byte beginnen. Danach folgen Filterdaten (4 Byte). Erst dann beginnen Geräusche und Musik.

Zeile 270 enthält die Filterdaten:

1. Byte – Filterfrequenz, Low-Byte

... bestimmt zusammen mit dem zweiten Byte die Frequenz, wenn ein Filter mit Byte 3 eingeschaltet und sein Modus mit und Byte 4 festgelegt ist.

2. Byte – Filterfrequenz, High-Byte

... bestimmt zusammen mit dem ersten Byte die Filterfrequenz.

3. Byte – Resonanz/Filter an/aus

... legt die Resonanz der Filter (0 bis 15 x 16) fest und schaltet die einzelnen Filter ein (0 bis 15).

4. Byte – Filtermodus/Lautstärke

... setzt den Filtermodus (0 bis 15 x 16) und die Lautstärke aller drei Stimmen (0 bis 15).

Eine genauere Beschreibung der SID-Register-Funktionen finden Sie ab S. 35 im Sonderheft 53.

Ab Zeile 310 befindet sich die Daten für Geräusche uund Musik. Das erste Byte ist jeweils der ASCII-Code für die Note. Interpretiert werden:

<C> = C
<SHIFT C> = Cis
<D> = D
<SHIFT D> = Dis
<E> = E
<F> = F
<SHIFT F> = Fis
<G> = G
<SHIFT G> = Gis
<A> = A
<SHIFT A> = Ais
<H> = H

Im zweiten Byte werden Note an (1) oder aus (0), Datensatznummer (0 bis 3), Oktave (0 bis 6) und Nummer der Stimme (0 bis 3) miteinander verknüpft:

Note an/aus (Bit 7)
Datensatznummer (Bit 5 und Bit 6)
Oktave (Bit 2 bis Bit 4)
Stimmen-Nummer (Bit 0 und Bit 1)

Damit dieses zweite Byte nicht für jede Note von Hand berechnet werden muß, lassen sich die Daten nebeneinander eintragen. Beachten Sie dazu die »REM«-Zeile 290, sie zeigt die Reihenfolge der Daten in den DATA-Zeilen.

Die Töne werden innerhalb eines Interrups solange aufgerufen, bis anstelle der Note ein »W« (Wait) erscheint. Der Bytewert dahinter gibt an, wie viele Interrupt-Zyklen bis zum nächsten Ton gewartet wird (ein Zyklus entspricht 1/60 Sekunde).

Achtung: In einem »W«-Zyklus lassen sich zwar beliebig viele Noten einbauen, dieses Vorgehen ist aber nicht sinnvoll, weil jede Note eine gewisse Bearbeitungszeit benötigt. Sind zuviele Noten für einen Zyklus eingebaut, wird die zulässige Zeit für einen IRQ-Zyklus überschritten. Außerdem hören Sie bei einer Änderung einer Stimme innerhalb eines Zyklus nur den letzten (geänderten) Ton. Daher ist es sinnvoll, max. alle drei Stimmen (wie auch von Zeile 310 bis Zeile 330) mit Tönen zu belegen. Danach setzen Sie das »W« (Zeile 340).

Effektdaten müssen am Anfang der Daten stehen und mit »S« (Stop) abgeschlossen sein. Es sind max. vier Effekte erlaubt. In unserem Listing wurden zwei Effekte (Zeile 310 bis 350 und Zeile 370 bis Zeile 430) verwendet. Danach folgt die eigentliche Musik (ab Zeile 450):

Das Musikstück muß als letzten Datensatz »R« (Repeat) enthalten (Zeile 1320). Trifft die Abspielroutine auf dieses »R«, beginnt Sie das Stück wieder ab dem letzten »S«.

Die Abspielroutine erkennt beim Intialisieren automatisch, wie viele Effekte programmiert sind. Dadurch wird ein Overlay mit neuen Daten möglich. Wenn Sie aus Basic neue Daten nachladen, müssen Sie allerdings vorher mit »SYS49152« ausschalten. Danach laden Sie die neuen Sounddaten und initialisieren neu (»SYS49152«).

Bei eigenen Spielen ist es meist nicht sinnvoll, die Musik in einem eigenen Interrupt laufen zu lassen. Dazu sind Spiele durch ihre komplexe Interrupt-Programmierung zu zeitkritisch. Wir werden später noch hören warum. In so einem Fall ist es besser die Soundroutine zuerst zu initialisieren und als letzten Teil des eigenen IRQ's direkt anzuspringen. Dazu müssen Sie folgendermaßen vorgehen:

Initialisieren Sie aus der Routine, die für Ihr Spiel den IRQ verbiegt mit

```
JSR $C100 ; initialisiert die Tabelle Geräusch/Musik
JSR $C0CC ; setzt die internen Pointer richtig
```

Danach müssen Sie die Rücksprungadresse zu Ihrem Programm, oder zum alten Interrupt setzen:

```
LDA #<RUECKSPRUNG
LDX #>RUECKSPRUNG
STA $C003
STX $C004
```

wobei »RUECKSPRUNG« die von Ihnen gewünschte Adresse darstellt. Aus Ihrer Interruptschleife verwenden Sie dann die Routine mit

```
JMP $C0A5
```

Sie springt nach der Abarbeitung zur Speicherposition »RUECKSPRUNG«.

Effekte rufen Sie aus Ihrem Programm mit

```
LDA #EFFEKTNUMMER
STA $C00D
```

auf. Beim nächsten Aufruf der Routine wird der normale Sound unterbrochen, und der Effekt eingeleitet.

Zum Nachladen von Tondaten müssen Sie sicherstellen, daß die Routine während der Ladeoperation nicht angesprungen wird. Danach ist es wichtig mit

```
JSR $C100
JSR $C0CC
```

neu zu initialisieren.

Kurzinfo: Musik

Programmart: Beispielprogramm für einen Effekt- und Musikdatensatz
Laden: LOAD"MUSIK",8
Starten: RUN
Besonderheiten: Endet mit »OUT OF DATA ERROR ?«
Benötigte Blocks: 12 Blocks
Programmautor: Herbert Großer

Kurzinfo: Sound

Programmart: Abspielroutine für Effekte und Musik
Laden: LOAD"SOUND",8,1
Starten: nach dem Laden NEW und SYS49152 eingeben
Benötigte Blocks: 4 Blocks
Programmautor: Herbert Großer

Geräusche und Hintergrundmusik

Überlegen wir uns grundsätzlich, wie wir diese Musik realisieren können. Es läßt sich z.B. ein Musikstück mit einem Sound-Editor erzeugen und wir binden die erzeugte Musikroutine später in unser Programm ein. Dieser Weg wird auch von den meisten kommerziellen Programmen gegangen. Es muß dabei allerdings berücksichtigt werden, wo diese Routine, in welcher Länge plaziert ist und wie sie initialisiert und eingebunden werden muß. Daher Vorsicht vor unbekannten Sound-Editoren: Die Analyse der Soundroutinen bedeutet eine Menge Arbeit und viel Programmier-Erfahrung. Doch hier liefern wir Ihnen eine Universalroutine mit, die Sie editieren, verändern und in eigene Programme einbinden können. Beschreibung und Bedienung finden Sie im Textkasten »die Soundroutine« auf S. 22.

Programmiergrundlagen

Wir haben jetzt die meisten kreativen Vorbereitungen beendet und Zeichensatz- Grafik- und Sprites auf Diskette gespeichert. Nun ist es an der Zeit, sich zu überlegen, wie programmiert werden soll. Dafür gibt es zwei grundsätzliche Methoden:

1. Die »ich fange an und programmiere los« Methode und
2. »modulare« Programmierung.

Mit der ersten Methode kommt man zwar bei kleinen Projekten zu einem Ergebnis, es wird aber selten so aussehen, wie man es sich vorgestellt hat. Abgesehen davon ist der Zeitaufwand nicht abschätzbar und ein Fehler läßt sich fast nicht nachvollziehen, da sich einzelne Funktionsblöcke nicht abgrenzen, sondern quer durchs Programm schlängeln.

Die zweite und bessere Methode, »modulares Programmieren«, soll bei unserem Kurs zur Anwendung kommen. »Modular« bedeutet in Modulen, also mit räumlich abgegrenzten Programmteilen zu arbeiten. Sie werden später über ein Hauptmodul miteinander verbunden und haben gegenüber der »Schlauchprogrammierung« entscheidende Vorteile:

1. Es kann jede einzelne Funktion für sich getestet werden bis sie ohne Einschränkung funktioniert.
2. Nach und nach entsteht eine Bibliothek, die nach Belieben für einen anderen Programmzweck zusammengestellt werden kann – denn warum soll das Rad jedesmal neu erfunden werden.
3. Sollte sich herausstellen (z.B. bei zeitkritischen Anwendungen) daß ein Modul zu langsam ist, kann dieses geändert oder ausgetauscht werden, ohne das übrige Programm zu beeinflussen.

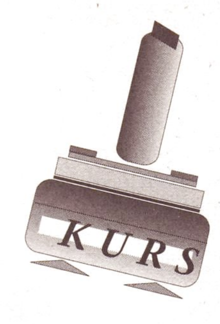
Für unser Spiel besitzen wir bereits ein Programm-Modul: Die Sound-Abspielroutine.

Überlegen wir uns anhand des Flußdiagramms (Abb. 1, Seite 16), welche Module wir weiterhin brauchen:

1. Wir steuern ein Fantasieraumschiff durch ein Labyrinth, natürlich mit dem Joystick. Also benötigen wir eine Routine, die unser Raumschiff steuert. Nun gibt es mehrere Möglichkeiten der Steuerung: Entweder wir bewegen das Sprite, oder den Hintergrund, oder beide. Das Sprite allein zu steuern, bringt nicht viel, da der Hintergrund größer als ein Bildschirm ist. Daher ist es besser, den Hintergrund zu bewegen (»scrollen«). Damit wir nach vorne und hinten ausweichen können, bewegen wir zusätzlich das Raumschiff-Sprite.
2. Dann lassen wir Gegner auftreten, sie müssen sich quasi automatisch bewegen.
3. Wir haben außerdem beschlossen, Türen einzubauen, die sich öffnen und schließen. Erreichen können wir dies durch Veränderung der Bitmuster des Zeichengenerators. Wenn man das Bitmuster eines Zeichens ändert, werden alle Zeichen des gleichen Codes am Bildschirm ebenfalls verändert.
4. Zudem haben wir beschlossen, unser Raumschiff ständig zu drehen. Dazu verwenden wir die »Animation«. Diesen optischen Trick kennen wir von Film und Fernsehen. Es werden nacheinander leicht veränderte Bilder gezeigt. Wenn dies schnell genug geschieht, entsteht der Eindruck einer fließenden Bewegung.
5. Ein wichtiges Modul fehlt uns noch:

Berührt unser Raumschiff einen Gegner oder den Hintergrund, muß dieses Ereignis festgehalten werden.

6. Schließlich und endlich soll alles miteinander zu einer Einheit verbunden, alle Module initialisiert, die einzelnen Ereignisse ausgewertet und darauf reagiert werden. Dieses sechste Modul stellt die eigentliche IRQ-Routine zur Verfügung, von der aus die anderen Module aufgerufen werden. Sie legt damit den Spielablauf fest.



Variablen

Im Gegensatz zur Programmierung in Basic verwalten wir in Assembler direkt die Werte in und aus Speicherstellen. Bei Variablen, die größer als 255 sind, verknüpfen wir mehrere davon. Zudem benötigen wir Pointer in der Zero-Page. Einige verwenden wir für indirekte Sprünge, andere ändern wir einfach und klinken uns so in schon bestehende Betriebssystem-Routinen ein.

Bei der Entwicklung der sechs Module ist einerseits darauf zu achten, daß die verwendeten Pointer und Variablen nicht von mehreren Routinen belegt und geändert werden, sehr seltsame Dinge geschähen sonst. Andrerseits müssen einige Parameterübergaben zwischen den einzelnen Programmteilen ermöglicht werden. Beispielsweise ist die Information »es fand eine Berührung Sprite/Hintergrund statt« unbedingt notwendig, sonst könnte nicht darauf reagiert werden:

1. Wir reservieren einen Bereich des Speichers speziell für alle Übergabe-Variablen und geben diesen im Assembler-Quelltext unverwechselbare Namen.
2. Variable, die nur im Modul verwendet werden, legen wir auch in diesen Speicherbereich. Dann ist es fast ausgeschlossen, daß einzelne Variablen kollidieren.
3. Wenn wir Pointer in der Zero-Page verwenden, weisen wir jeder Routine andere Speicherstellen zu. Eine andere Methode wäre es, die alten Pointerwerte zwischenzuspeichern, dann im Modul neu zu belegen und zum Schluß die vorher gespeicherten Werte wieder in den Pointer zu übertragen. Es spielt dann keine Rolle mehr, ob der Pointer auch in einer anderen Routine verwendet wird. Leider ist dieser Weg bei Spielen nicht möglich, da die meisten Routinen zeitkritisch sind und jeder unnötige Zeitzyklus vermieden werden sollte.

Im Textkasten »Zero-Page Adressen« (S. 21) sehen Sie eine Übersicht der Pointer, die das Betriebssystem nicht behindern.

Flußdiagramme

Wir haben eine wichtige Vorbereitung zur Programmierung schon gesehen – das Flußdiagramm. Für unsere Spielidee genügte es auch, einfach über ein paar Kästchen die Gedankengänge schriftlich zu fixieren. Wenn wir jetzt aber ans »Ein-

Quickreferenz SMON. Die Klammern dürfen nicht mit eingegeben werden. Die Werte in den Klammern können, aber müssen nicht eingegeben werden.

Befehl	Funktion
A 4000	Zeilenassembler Startadresse = $4000.
B 4000 4200	Erzeugt Basic-DATA-Zeilen im Bereich $4000 bis $41FF
C 4010 4200 4013	
4000 4200	Verschieben eines Programmes mit Adreßumrechnung. Entspricht W- und V-Befehl.
D 4000 (4100)	Disassembliert den Bereich von $4000 bis $4100
F	findet Zeichenketten (F), absolute Adressen (FA), relative Sprünge (FR), Tabellen (FT), Zeropage-Adressen (FZ) und Immediate-Befehle (FI).
GO 4000	Startet Maschinenprogramm (ab $4000)
IO 1	Ein-/Ausgabegerät auf Datasette umstellen
K A000 (A100)	Im angegebenen Bereich nach ASCII-Zeichen suchen.
L"name" (4000)	Laden eines Programmes an die richtige (oder angegebene) Adresse
M 4000 (4100)	Gibt den Inhalt des angegebenen Speicherbereichs als Hex-Byte und ASCII-Zeichen aus.
O 4000 4100 12	Füllt den angegebenen Bereich mit $12.
PO 5	Setzt Drucker-Geräteadresse auf 5
R	Registerinhalte anzeigen
S"name" 4000 4500	
	Speichert ein Programm von $4000 bis $4FFF.
TW (4000)	Einzelschrittmodus. Mit »J« können Unterprogramme in Echtzeit ausgeführt werden.
TB 4010 (05)	Breakpoint setzen (nach dem 5. Durchlauf)
TQ 4000	Schnellschrittmodus. Springt beim Erreichen eines Breakpoints in die Registeranzeige.
TS 4000 4020	Arbeitet ein Programm ab $4000 in Echtzeit ab und springt beim Erreichen von $4020 in die Registeranzeige.
V 6000 6200 4000 4100 4200	
	Ändert alle absoluten Adressen $4000 bis $41FF, die sich auf den Bereich $6000 bis $6200 beziehen, auf den neuen Bereich $4000.
W 4000 4300 5000	Verschiebt den Speicherinhalt von $4000 bis $42FF nach $5000.

Befehl	Funktion
X	Monitor verlassen.
# 49152	Dezimal umrechnen
$ 002B	Vierstellige Hex-Zahl umrechnen.
% 01100100	Achtstellige Binärzahl umrechnen.
? 0344+5234	Addition oder Subtraktion zweier vierstelliger Hex-Zahlen.
= 4000 5000	Vergleicht den Speicherinhalt von $4000 bis $5000.
Z	Ruft den Diskettenmonitor auf (falls implementiert). Dieser verfügt über folgende Befehle
R (12 01)	Liest Track $12 Sektor $01. Fehlt die Angabe hinter »R«, wird der logisch nächste Sektor gelesen.
W (12 01)	Schreibt Track $12 Sektor $01 auf Diskette. Fehlt die Angabe hinter »W«, werden die letzten Angaben von »R« benutzt.
M	Zeigt den Pufferinhalt als Hex-Dump.
X	Rücksprung zum Monitor.
F	Weitere Diskettenbefehle initialisieren (falls implementiert). Sind die Befehle initialisiert, stehen folgende Befehle zur Verfügung.
M (07)	Memory-Dump (Floppy-RAM/ROM) ausgeben.
V 6000 0400	Verschiebt einen 256-Byte-Block von $6000 ins Floppy-RAM nach $400.
@	Normale Diskettenbefehle senden
X	Zurück in normalen Diskettenmonitor.
Ist die Erweiterung »Neues vom SMON« implementiert, stehen folgende Befehle zur Verfügung:	
Z 4000 (4100)	Gibt den Speicherinhalt von $4000 bis $40FF binär aus (ein Byte pro Zeile).
H 4000 (4100)	Gibt den Speicherbereich von $4000 bis $40FF binär aus (drei Byte pro Zeile).
N 4000 (4100)	Gibt den Speicherinhalt von $4000 bis $40FF im Bildschirmcode aus (32 Zeichen pro Zeile).
U 4000 (4100)	Wie »N« aber 40 Zeichen pro Zeile. Änderungen sind nicht möglich.
E 4000 (4100)	Füllt den Speicherbereich von $4000 bis $40FF mit $00.
Y 40	Verschiebt den SMON nach $4000.
Q 2000	Kopiert den Zeichensatz nach $2000.
J	Bringt letzten Ausgabebefehl zurück.

gemachte« gehen und das erste Modul vom Flußdiagramm bis zur Routine entwickeln, sollten wir uns ansehen, warum die einzelnen Kästchen unterschiedliche Formen haben:

Kurzinfo: Symbole nach DIN 66001 für Flußdiagramme

Beginn oder Ende eines Programms (oder Programmteils)

Allgemeine Operation (es muß vom Computer etwas durchgeführt werden)

Verzweigung in eine von zwei Richtungen

Übergangs- oder Schnittstelle

Unterprogramm

Selbstmodifizierung von Programmen

Operation, die nicht vom Computer durchgeführt wird (z.B. Tastendruck)

Eingabe- oder Ausgabeoperationen (z.B. Eingaben für Berechnungen)

Ablauflinie

Zusammenführung von Ablauflinien

Bemerkungen zum Ablaufplan

Ein Flußdiagramm ist eine schnelle und übersichtliche Methode bei der Entwicklung eigener Routinen, wenn Sie drei Dinge beachten:

1. Register und Pointer sollten grundsätzlich als erstes (außerhalb der Symbole) notiert werden. Ohne ihre Kenntnis stricken Sie das Flußdiagramm mehrere Male um.

2. Versuchen Sie nicht, alle Details in die Symbole einzutragen sonst leidet die Übersicht.

3. Bei zu wenig Information haben Sie keine Chance, den Ablauf nachzuvollziehen.

Beginnen wir mit Modul 1, der Joystickabfrage und Überlegen wir uns zuerst, welche Register wir dazu benötigen (Tabelle 1, S. 26 und Tabelle 3, S. 27):

Wir verwenden Joyport 2 für unsere Abfrage. Das entsprechende Register ($DC00) dafür muß zuerst vorbereitet werden, da es vom Betriebssystem zur Spaltenauswahl bei der Tastaturabfrage verwendet wird, also auf Ausgabe geschaltet ist. Wir schalten um, indem wir in Register $DC02 den Wert $E0 schreiben. Dann läßt sich in Register $DC00 das Bit-Muster des Joyport 2 auslesen. Dabei entsprechen fünf Bits jeweils einer Joystick-Funktion:

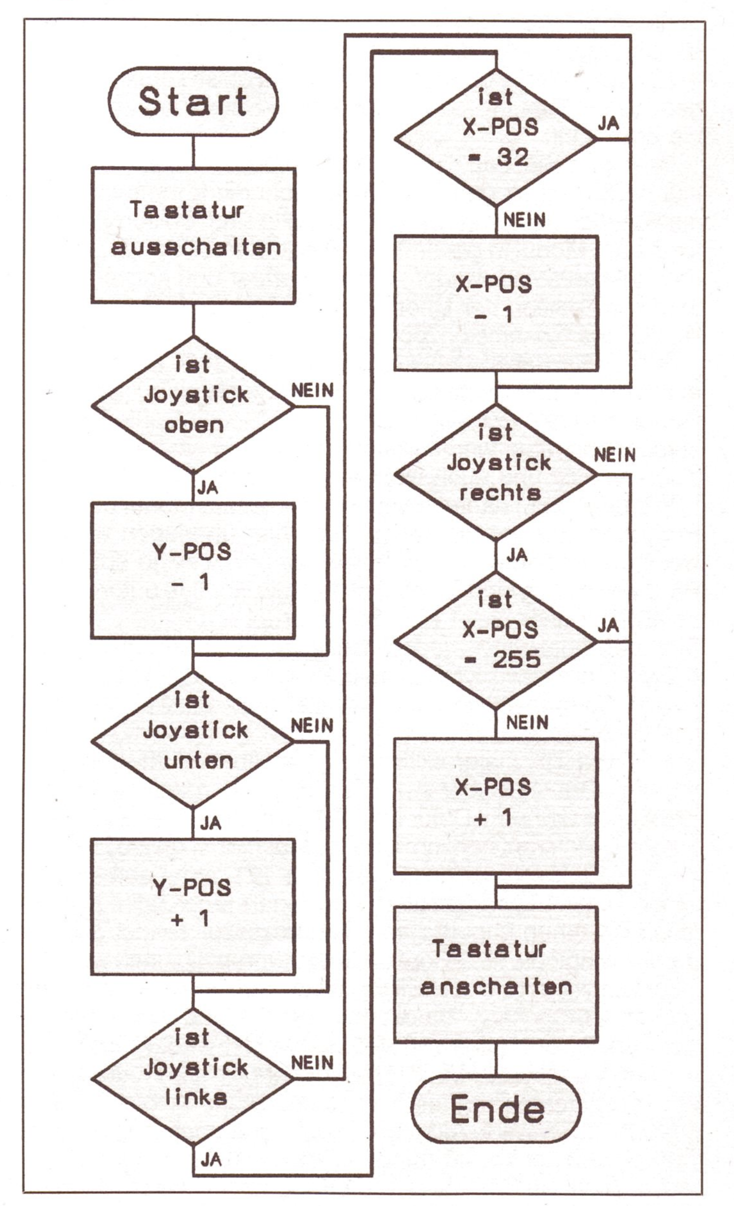
Bit 0 = oben
Bit 1 = unten
Bit 2 = links
Bit 3 = rechts
Bit 4 = Knopf (Feuer)

Achtung: Der Ausgangswert bei unbetätigtem Joystick ist $FF, nicht $00, wie man meinen könnte. Der Grund liegt im Prinzip des Joysticks. Für jede Position (oben, unten, links, rechts und Knopf) ist ein eigener Taster im Joystick eingebaut. Eine Betätigung verbindet die entsprechende Portleitung mit Masse (low). Es muß also eine Spannung an den Portleitungen anliegen (+5V). Im Computer ist grundsätzlich vereinbart, daß Spannung (+5V) als High, also gesetztes Bit interpretiert wird. Eine Joystick-Aktion löscht daher das gesetzte Bit. Wir prüfen das mit einer AND-Maske.

Legen wir noch fest, daß unser Raumschiff Sprite 0 ist. Danach notieren wir die nötigen Sprite-Register (s. Tabelle 3, S. 28) und die verwendeten Pointer:

Datenrichtungsregister = $DC02
JOYPORT = $DC00
V = $D000 (Basisadresse Video-Interface-Chip)
x-Position/Sprite 0 = V
y-Position/Sprite 0 = V+1

Aus diesen Grundlagen heraus überlegen wir uns ein Flußdiagramm:



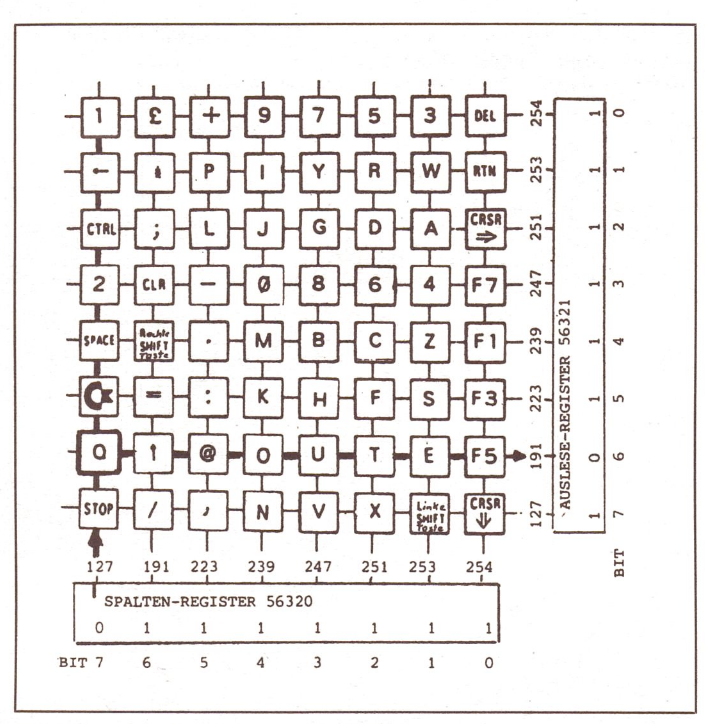


Tabelle 1. Die Tastaturmatrix im C64

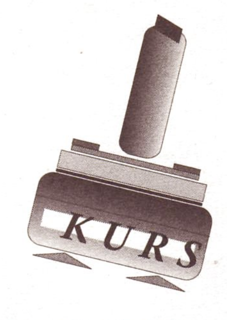
Wenn wir nun unser Maschinenprogramm für die Joystickabfrage entwickeln, müssen wir die Aktionen nur noch in Assemblerbefehle umsetzen. Sie finden diese Routine im abgedruckten Gesamt-Listing des Demo-Spiels von Zeile 1100 bis Zeile 1360.

Beachten Sie, daß ein Quelltext immer kommentiert in die eigene Bibliothek gehen sollte, obwohl die Kommentare für die Funktion des Programms nicht nötig sind. Sollten Sie später dieses Modul in ein anderes Programm einbauen, ist seine Funktion sofort wieder nachvollziehbar und kritische Partien, wie Position der Übergabe-Variablen usw., lassen sich leichter ins Gesamtkonzept einplanen.

Sollten Sie das Listing aus Zeile 1100 bis 1360 abtippen, assemblieren und testen, werden Sie feststellen, daß sich (scheinbar) rein gar nichts tut. Warum? - das Sprite ist zwar verschoben, aber weder definiert, noch auf eine Anfangsposition gesetzt und auch nicht eingeschaltet.

Wir benötigen daher zusätzlich zu diesem Modul noch ein Programm zur Initialisierung. Auch hier überlegen wir uns, welche Funktionen in der Testphase und welche später im Programm nötig sind: Zunächst muß die Routine automatisch ablaufen, daher bietet sich der Interrupt an:

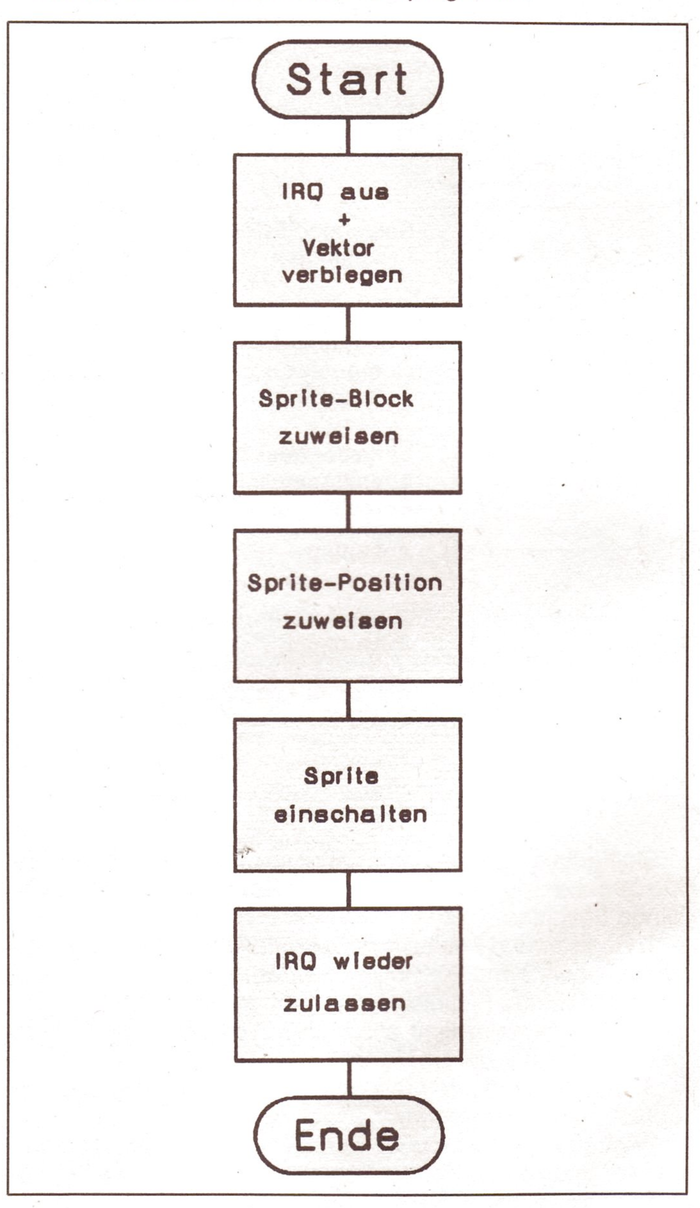
Sie errinnern sich, jede sechzigstel Sekunde unterbricht der Mikroprozessor das normale Programm und tut alles das, was im Computer scheinbar von allein abläuft. Dies geht vom Cursorblinken über die Tastaturabfrage, bis hin zur Erhöhung von TI und TI$. Dafür existiert im Betriebssystem eine extra Interrupt-Routine. Aber ein IRQ muß auch irgendwie ausgelöst werden können. Dafür ist CIA 1 (Complex-Interface-Adapter) im C64 verantwortlich. Dieser Baustein ist gleich zweimal im Computer vorhanden (Tabelle 2, S. 27). Jeder enthält zwei 16-Bit-Timer (Abwärtszähler) von denen jeder beim Zählerstand »0« einen Impuls zum Mikroprozessor sendet und ihn damit zwingt, die IRQ-Routine auszuführen. Danach wird der Baustein wieder mit dem voreingestellten Wert geladen und erneut herabgezählt. Da die Interrupt-Routine über einen indirekten Sprung (JMP ($0314)) verwendet wird, müssen wir nur den Vektor ($0314/$0315) auf unsere eigene Routine richten. Im späteren Programm ist genau dieses Verbiegen nicht erlaubt, da andere Aufgaben auch »automatisch« erledigt werden und der Vektor immer auf die zuletzt verbogene Routine zeigen würde. Alle anderen Routinen wären dadurch



ausgeschaltet. Wir werden daher später alle Module zusammenfassen und der Reihe nach aufrufen. Außerdem lernen wir noch andere Arten kennen, die den Interrupt auslösen. Zum Schluß unserer Testroutine springen wir in die Interrupt-Routine des Betriebssystems. Auch diesen Teil benötigen wir im Spiel nicht.

Bevor wir wieder ein Flußdiagramm aufzeichnen, notieren wir uns die benötigten Register und Pointer:
IRQ-Vektor - $0314/$0315
Sprite-Block - 32
Sprite-Pointer - 2040
Sprite 0 - 1 (#00000001)
Sprite Ein - VIC+21

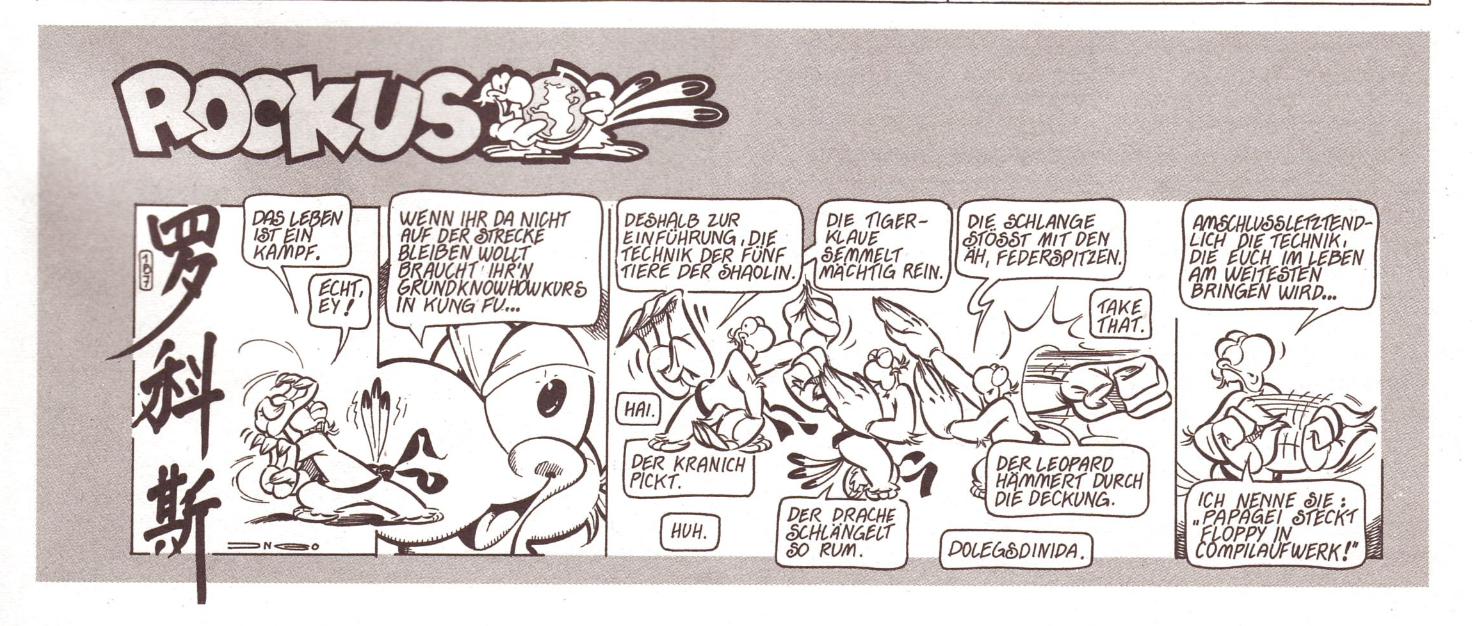
Daraus entsteht dann unser Testprogramm:

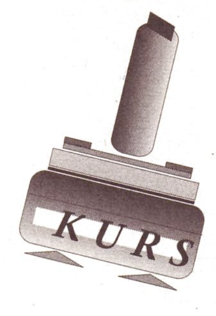


Mit ihm lassen sich alle Funktionen überprüfen. Im Demo-Spiel ist die Routine »SETSPRITE« ab Zeile 150 dafür zuständig, daß Sprite 0 auf x-Position 130, y-Position 114 und auf Normalgröße gesetzt wird. Weiterhin wird das Sprite auf Multicolor, mit den Farben Rot und Grün gesetzt. Die Daten stehen in Sprite-Block 32.

Reg.	Adresse		Funktion	Verwendung
0	56320 56576	($dc00) ($dd00)	Datenregister Port A Ein gesetztes Bit signalisiert High an der entsprechenden Port-Leitung	Tastaturabfrage IEC-Bus + RS232
1	56321 56577	($dc01) ($dd01)	Datenregister Port B Wie Register 0, jedoch für Port B	Tastaturabfrage User-Port
2	56322 56578	($dc02) ($dd02)	Datenrichtungsregister Port A Ein gesetztes Bit programmiert die zugehörige Portleitung als Ausgang	zusammen mit Register 0 zusammen mit Register 0
3	56323 56579	($dc03) ($dd03)	Datenrichtungsregister Port B Wie Register, jedoch für Port B	zusammen mit Register 1 zusammen mit Register 1
4	56324 56580	($dc04) ($dd04)	Timer A, Low-Byte Beim Lesen wird der momentane Zählerstand erhalten, beim Schreiben der Zählerstand (Low-Byte) gesetzt, von dem der 16-Bit-Zähler nach Null zählt	IRQ (alle $\frac{1}{60}$ s) RS232
5	56325 56581	($dc05) ($dd05)	Timer A, High Byte Wie Register 4, jedoch für High-Byte, Timer A Siehe auch Register 14 (Control-Register A)	zusammen mit Register 4 zusammen mit Register 4
6	56326 56582	($dc06) ($dd06)	Timer B, Low-Byte Wie Register 4, jedoch für Timer B Siehe auch Register 15 (Control-Register B)	für Kassetten Op. RS232
7	56327 56583	($dc07) ($dd07)	Timer B, High-Byte Wie Register 5, jedoch für Timer B Siehe auch Register 15 (Control-Register B)	zusammen mit Register 6 zusammen mit Register 6
8	56328 56584	($dc08) ($dd08)	Time of Day $\frac{1}{10}$ Sekunden Bit 0-3 enthalten die $\frac{1}{10}$ Sekunden im BCD-Format. Ist Bit 7 in Register 15 gesetzt, so wird beim Schreiben die Alarmzeit gesetzt, ansonsten die Uhrzeit. Bit 4-7 unbenutzt.	(für RND) unbenutzt
9	56329 56585	($dc09) ($dd09)	Time of Day Sekunden Dieses Register enthält die Sekunden im BCD-Format. Schreibzugriff siehe Register 8	(für RND) unbenutzt
10	56330 56586	($dc0a) ($dd0a)	Time of Day Minuten Dieses Register enthält die Minuten im BCD-Format Schreibzugriff siehe Register 8	(für RND) unbenutzt
11	56331 56587	($dc0b) ($dd0b)	Time of Day Stunden Bit 0-3 enthalten die Stunden im BCD-Format, Bit 4 die 10er Stunden. Bit 7 ist bei AM (vormittags) 0 und bei PM (nachmittags) 1. Bit 5+6 unbenutzt Schreibzugriff siehe Register 8	(für RND) unbenutzt
12	56332 56588	($dc0c) ($dd0c)	Serial Data Register (SDR) Schieberegister, über das Daten am Pin SP herausgeschoben und hereingeholt werden. Das höchstwertige Bit erscheint zuerst.	unbenutzt unbenutzt
13	56333 56589	($dc0d) ($dd0d)	Interrupt Control Register (ICR) Bit 0: Unterlauf Timer A Bit 1: Unterlauf Timer B Bit 2: Uhrzeit und Alarmzeit sind gleich Bit 3: Schieberegister voll oder leer (je nach Betriebsart) Bit 4: 1, wenn negative Spannungsflanke an FLAG aufgetreten ist Bit 5 und Bit 6 sind immer 0 Bild 7: Es stimmt mindestens ein gesetztes Bit im INT MASK und INT DATA-Register überein Achtung: Beim Lesen wird das ICR gelöscht!	
14	56334 56590	($dc0e) ($dd0e)	Control Register A (CRA) Bit 0: 1=Timer A starten 0=Timer A stoppen Bit 1: 1=Ein Umlauf von Timer A wird an PB 6 signalisiert, auch wenn dieses Port-Bit als Eingang programmiert ist. Bit 2: 1=Bei einem Unterlauf von Timer A wird PB 6 invertiert Bit 3: 0=Continous-Mode 1=One-Shot-Mode Bit 4: Wird eine 1 eingeschrieben, so wird Timer A sofort mit dem Wert geladen, der vorher in Register 4 + 5 stand, egal ob der Timer gerade läuft oder nicht. Bit 5: 1=Timer A zählt positive Flanken an CNT 0=Timer A zählt Systemtakte Bit 6: 0=Das Schieberegister ist Eingang 1=Das Schieberegister ist Ausgang Bit 7: 1=TOD verarbeitet 50 Hz Netzfrequenz 0=TOD verarbeitet 60 Hz Netzfrequenz	
15	56335 56591	($dc0f) ($dd07)	Control Register B (CRB) Bit 0-4: entsprechen Bit 0-4 von CRA, jedoch für Timer B und PB 7 Bit 5+6 bestimmen paarweise die Triggerquelle 00=Timer B zählt Systemtakte 01=Timer B zählt positive Flanken an CNT 10=Timer B zählt Unterläufe von Timer A 11=Timer B zählt Unterläufe von Timer A nur, wenn CNT high ist Bit 7: 1=TOD Alarmzeit setzen 0=TOD Uhrzeit setzen	

Tabelle 2. Register der CIAS





Die Register des VIC
Basisadresse 53248 ($D000)

Register	Normwert	Zweck
0 53248 $D000	0, wenn Sprite nicht verwendet	x-Koordinate von Sprite #0 ist der Wert < 24, dann ist der Sprite verdeckt.
1 53249 $D001	0, wenn Sprite nicht verwendet	y-Koordinate von Sprite #0 ist der Wert < 50 > 229, dann ist der Sprite verdeckt
2 53250 $D002	siehe 0	x-Koordinate von Sprite #1 sonst siehe 0
3 53251 $D003	siehe 0	y-Koordinate von Sprite #1 sonst siehe 0
4 53252 $D004	siehe 0	x-Koordinate von Sprite #2 sonst siehe 0
5 53253 $D005	siehe 0	y-Koordinate von Sprite #2 sonst siehe 0
6 53254 $D006	siehe 0	x-Koordinate von Sprite #3 sonst siehe 0
7 53255 $D007	siehe 0	y-Koordinate von Sprite #3 sonst siehe 0
8 53256 $D008	siehe 0	x-Koordinate von Sprite #4 sonst siehe 0
9 53257 $D009	siehe 0	y-Koordinate von Sprite #4 sonst siehe 0
10 53258 $D00A	siehe 0	x-Koordinate von Sprite #5 sonst siehe 0
11 53259 $D00B	siehe 0	y-Koordinate von Sprite #5 sonst siehe 0
12 53260 $D00C	siehe 0	x-Koordinate von Sprite #6 sonst siehe 0
13 53261 $D00D	siehe 0	y-Koordinate von Sprite #6 sonst siehe 0
14 53262 $D00E	siehe 0	x-Koordinate von Sprite #7 sonst siehe 0
15 53263 $D00F	siehe 0	y-Koordinate von Sprite #7 sonst siehe 0
16 53264 $D010	siehe 0	9. Bit der x-Koordinaten für Pos. 256-319 zuständig: Bit: 7|6|5|4|3|2|1|0 Sprite: 7|6|5|4|3|2|1|0
17 53265 $D011	27	Kontrollregister für Modus des VIC Belegung: 0, 1, 2 } weiches scrollen (0-7) 3 25/24 Zeilen 1/0 4 VIC ein/aus 1/0 5 HiRes/Text 1/0 6 Hintergrundfarb/Text 1/0 7 9.Bit Rasterwert
18 53266 $D012	wechselt	Nummer der gerade aktuellen Rasterzeile
19 53267 $D013	wechselt	x-Koordinate des Lightpen
20 53268 $D014	wechselt	y-Koordinate des Lightpen
21 53269 $D015	0, wenn Sprite nicht verwendet	Schaltet die Sprites aus/ein aus (0) / ein (0) Bit: 7|6|5|4|3|2|1|0 Sprite: 7|6|5|4|3|2|1|0
22 53270 $D016	200	Kontrollregister für erweiterte Funktionen 0, 1, 2 } weiches scrollen (0-7) 3 40/38 Textspalten 1/0 4 Multicolor/Normal 1/0 5 immer 0 6 immer 1 7 immer 1

Die Register des VIC
Basisadresse 53248 ($D000)

Register	Normwert	Zweck
23 53271 $D017	0, wenn Sprite nicht verwendet	x-Vergrößerung Sprite # 0 klein (0) / groß (1) Bit: 7|6|5|4|3|2|1|0 Sprite: 7|6|5|4|3|2|1|0
24 53272 $D018	21	Speicheraufteilung des VIC 0 immer 1 1, 2, 3 } Character-RAM/ROM 4, 5, 6, 7 } Bildschirm-RAM
25 53273 $D019	wechselt	IRQ Statusregister Ergebniss eines IRQ-Ereignis 0 Rasterzeile 1 Sprite-Hintergrundkollision 2 Sprite-Spritekollision 3 Lightpen 4 immer 1 5 immer 1 6 immer 1 7 Interrupt stattgefunden(1)
26 53274 $D01A	wechselt	IRQ Maske Die in Reg. 25 abzufragenden Ereignisse werden mit Bit=1 zugelassen 0 Raster-IRQ 1 Sprite-Hintergrund 2 Sprite-Sprite 3 Lightpen 4 immer 1 5 immer 1 6 immer 1 7 immer 1
27 53275 $D01B	0, wenn Sprite nicht verwendet	Priorität des Sprite 0 = vor, 1 = hinter dem Text Bit: 7|6|5|4|3|2|1|0 Sprite: 7|6|5|4|3|2|1|0
28 53276 $D01C	0, wenn Sprite nicht verwendet	Multicolor für Sprites 0 Schaltet aus, 1 schaltet ein Bit: 7|6|5|4|3|2|1|0 Sprite: 7|6|5|4|3|2|1|0
29 53277 $D01D	0, wenn Sprite nicht verwendet	y-Vergrößerung Sprite # 0 klein (0) / groß (1) Bit: 7|6|5|4|3|2|1|0 Sprite: 7|6|5|4|3|2|1|0
30 53278 $D01E	0, wenn Sprite nicht verwendet	Kollisionsregister Sprite-Sprite Bit= 0 wenn keine Kollision Bit: 7|6|5|4|3|2|1|0 Sprite: 7|6|5|4|3|2|1|0
31 53279 $D01F	0, wenn Sprite nicht verwendet	Kollisionsregister Sprite-Hintergrund Bit= 0 wenn keine Kollision Bit: 7|6|5|4|3|2|1|0 Sprite: 7|6|5|4|3|2|1|0
32 53280 $D020	254	Rahmenfarbe 0 = schwarz, 8 = orange 1 = weiß, 9 = braun 2 = rot, 10 = hellrot 3 = cyan, 11 = grau III 4 = rosa, 12 = grau II 5 = grün, 13 = hellgrün 6 = blau, 14 = hellblau 7 = gelb, 15 = grau I

Die Register des VIC Basisadresse 53248 ($D000)		
Register	Normwert	Zweck
33 53281 $D021	246	Bildschirmfarbe (Farben s. Reg. 32)
34 53282 $D022	241	Farbe 1 für Multicolor und erweiterter Hintergrundfarb-Modus (nur Text) (Farben s. Reg. 32)
35 53283 $D023	242	Farbe 2 für Multicolor und erweiterter Hintergrundfarb-Modus (nur Text) (Farben s. Reg. 32)
36 53284 $D024	243	Farbe 3 für Multicolor und erweiterter Hintergrundfarb-Modus (nur Text) (Farben s. Reg. 32)
37 53285 $D025	244	Farbe 1 für Multicolor-Sprite (Farben s. Reg. 32)
38 53286 $D026	244	Farbe 2 für Multicolor-Sprite (Farben s. Reg. 32)
39 53287 $D027	241	Farbregister Sprite #0 (Farben s. Reg. 32)
40 $D028 53288	242	Farbregister Sprite #1 (Farben s. Reg. 32)
41 $D029 53289	243	Farbregister Sprite #2 (Farben s. Reg. 32)
42 $D02A 53290	244	Farbregister Sprite #3 (Farben s. Reg. 32)
43 $D02B 53291	245	Farbregister Sprite #4 (Farben s. Reg. 32)
44 $D02C 53292	246	Farbregister Sprite #5 (Farben s. Reg. 32)
45 $D02E 53293	247	Farbregister Sprite #6 (Farben s. Reg. 32)
46 $D02F 53294	252	Farbregister Sprite #7 (Farben s. Reg. 32)

Tabelle 3. Alle Register des VIC

Mit unserer Routine läßt sich das Sprite in den von uns gesetzten Grenzen am Bildschirm bewegen. Aber wir haben noch keinen scrollenden Hintergrund. Überlegen wir uns zuerst eine Methode:

Beginnen wir bei der Initalisierung des Spiels und kopieren den ersten Teil des Landschaftsausschnitts in den Bildschirmspeicher. Danach warten wir auf den Spielebeginn (Druck auf den Feuerknopf). Anschließend kopieren wir pro Interrupt jeweils einen um ein Zeichen weiter rechts liegenden Ausschnitt aus unserer Spielelandschaft und erreichen damit ein sog. Hardscrolling. Bei der horizontalen Länge von 512 Zeichen wären wir nach 512-40=472 IRQ-Zyklen am Ende des Spielfelds angekommen. Das sind ca. acht Sekunden (472/60). Sie werden zugeben, zu schnell um im Spiel auf irgend etwas reagieren zu können. Also bauen wir zusätzlich einen Verzögerungszähler ein. Starteten wir jetzt unsere theoretische Routine, würde folgendes passieren:

1. Sowohl Sprite als auch Landschaft flackern hin und wieder.

2. Bei einer passablen Geschwindigkeit ruckelt unsere Landschaft unter dem Sprite.

Untersuchen wir das erste Phänomen genauer:

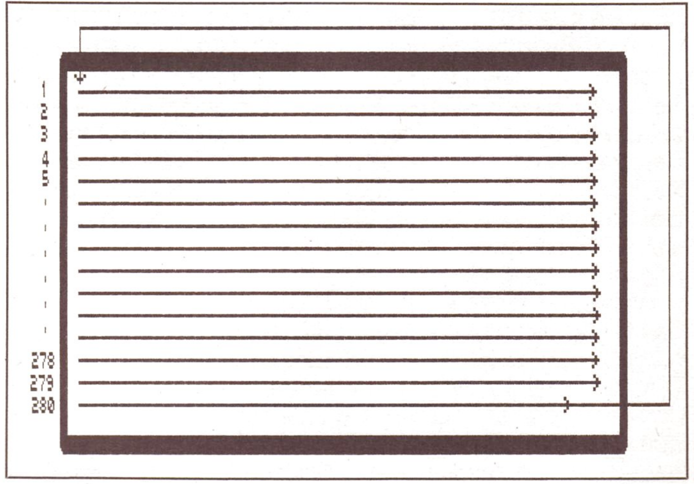
Der Grund für das Flackern ist der physikalische Aufbau eines Bildes. Er dauert 1/50stel Sekunde, findet also 50 mal pro Sekunde statt. Dabei beginnt ein Rasterstrahl (Abb. 4) links oben in der ersten Zeile am Monitor seine Arbeit. Je nachdem was er darstellen soll, leuchtet er heller oder dunkler, bzw. in einer anderen Farbe. Ist er am Ende der Zeile angekommen, beginnt seine Tätigkeit am Anfang der nächsten Zeile wieder. Da dies für 280 Zeilen innerhalb 1/50stel Sekunde passiert, wird unser Auge überlistet und für uns entsteht der Eindruck eines stehenden Bildes.

Wenn wir nun zu einem willkürlichen Zeitpunkt ein Muster am Bildschirm ändern, kann es passieren, daß ein Teil des Musters geändert wird, während der Rasterstrahl am Monitor gerade dieses Muster überholt. Bei diesem Bildzyklus sieht man also ein Teil des alten Musters und einen Teil des neuen. Da dies aber öfters geschieht, empfindet unser Auge ein deutliches Flackern. Abhilfe hätten wir, wenn es uns gelingt, die Änderung eines Bildschirminhalts mit dem physikalischen Bildaufbau zu synchronisieren. Und tatsächlich, der VIC bietet uns hier eine Möglichkeit, da er festhält, in welcher Zeile sich der Rasterstrahl gerade befindet. Zuständig dafür sind Register 53266 (V+18) und da es 280 Zeilen gibt, dient das Bit 7 von 53265 (V+17) als neuntes Bit (Abb. 5, S. 30). Wenn wir hier einen Wert vorgeben ist der VIC in der Lage, einen Interrupt auszulösen. Allerdings müssen wir dem VIC noch mitteilen, daß er einen IRQ durchführen soll. Dafür ist Bit 0 des Registers 53274 (V+26) zuständig. Dazu ein Beispiel; in ihm wird zwischen Groß/Kleinschrift und Groß/Grafik umgeschaltet:

In der Initialisierungs-Routine wird der IRQ-Vektor verbogen und danach die erste Rasterzeile für den IRQ festgelegt. Anschließend wird der IRQ durch Rasterzeilen zugelassen. Die neue IRQ-Routine löscht zuerst das Rasterregister und prüft gleichzeitig, ob der IRQ tatsächlich durch den VIC ausgelöst wurde (Negative-Flag). Die Rasterzeilen-Routine wechselt den Zeichensatz und legt die Rasterzeile neu fest, bei der ein IRQ ausgeführt wird.

Auch mit unserem zweiten Ärgernis, dem Ruckeln beim Scrollen, werden wir fertig - der VIC bietet auf einfachste Weise eine Soft-Scroll-Funktion:

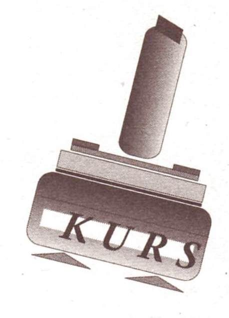
Ein Bildschirmzeichen besteht aus einer Matrix von achtmal acht Punkten. Wir sind bis jetzt davon ausgegangen, daß auch mit dem ersten Punkt des ersten Zeichens das Bild beginnt. Doch der VIC kann auch anders. Dazu bietet er ein Register. Bits 0 bis 2 des Registers 53270 (V+22) geben an, bei



[4] Der Aufbau des Bildschirms durch den Rasterstrahl

welchem Punkt die Bildschirmdarstellung beginnt. Für normalen Beginn ist dieser Wert demzufolge »0«. Wir verschieben unseren Bildschirm nach links, daher beginnen wir mit »7«. Wir müssen also siebenmal diesen Teil des Registers herabzählen, bis es den Wert »0« hat (Zeile 2230 bis 2450 im Listing, S. 33). Gleichzeitig dazu können wir den nächsten Ausschnitt ins Bildschirm-RAM kopieren (Zeile 1370 bis 1640). Ein kontinuierliches, weiches Scrollen nach links ist der Erfolg.

Bevor Sie für ein eigenes Spiel eine Scroll-Routine entwickeln, müssen Sie allerdings beachten: Ihre Routinen dürfen nicht länger dauern als ein physikalischer Bildaufbau, da

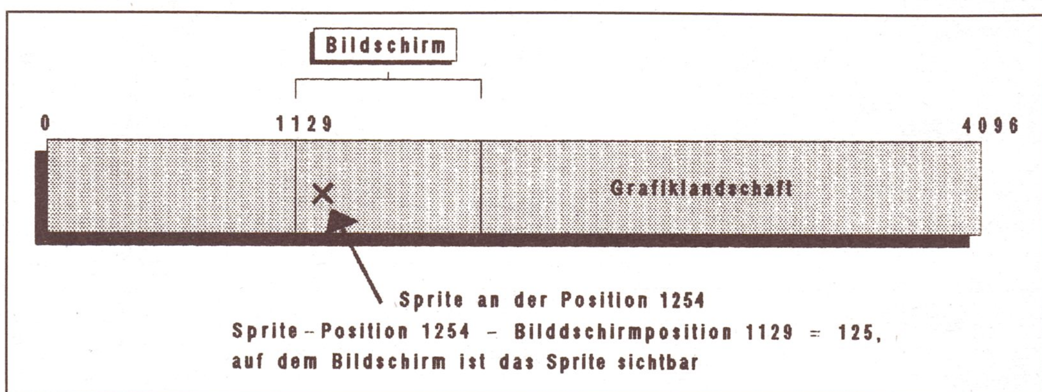


7 7 6 5 4 3 2 1 0

|8| |7|6|5|4|3|2|1|0|

VIC+17 VIC+18

[5] Das neunte Bit des Rasterstrahls ist Bit 7 im Register 17 des VIC. Zusammen mit Register 18 wird hier die Position eines Raster-IRQ definiert.



[6] Berechnung der Spriteposition

sich sonst das Bild selbst überholt. Es stehen Ihnen dafür während eines sichtbaren Rasterstrahldurchlaufs 200 x 63 = 12600 Taktzyklen zur Verfügung.

Automatische Gegner

Die konstruierte Grafik ist 512 Zeichen breit. Das bedeutet, sie hat 512 x 8 = 4096 Bildpunkte Ausdehnung. Über diese Breite darf sich ein computergesteuertes Hindernis (Sprite) bewegen. Es darf aber nur eingeschaltet sein, wenn es sich im momentan sichtbaren Bereich (320 Bildpunkte) der Spiele-Landschaft befindet. Daher muß berechnet werden, wann das Sprite sichtbar ist:

Wir zählen von Anfang an die aktuelle Position mit (»ZAEHLEN«, Zeile 4050 bis 4120) und reservieren dafür zwei Variablen »POSSCRLO« für das Low-Byte und »POSSCRHI« für das High-Byte. Im Unterprogramm »INITSPR« werden die Positionen der gegnerischen Sprites dann definiert. Jedem Sprite wird eine 16-Bit x-Position und 8-Bit y-Position zugeteilt; zusätzlich noch eine Speicherposition, in der die jeweilige Bewegungsrichtung gespeichert ist. Diese Daten stehen von Zeile 3730 bis 4000 im Listing. Es lassen sich durchaus mehr Gegner definieren, Sie müssen allerdings sicherstellen, daß sich nie mehr als acht Sprites gleichzeitig auf dem aktuellen Bildschirm tummeln können.

Während des Spielablaufs verfügen wir über die Position des Bildschirms und der gegnerischen Sprites. Daraus berechnen wir die jeweilige Spriteposition (Abb. 6):

Nehmen wir an, daß sich Sprite 1 an Position 1254 befindet und der Bildschirm auf Position 1129. Das Ergebnis der Rechnung Spriteposition (1254) minus Bildschirmposition (1129) ist genau die Spriteposition am Bildschirm (125). Ergibt die Rechnung einen Wert kleiner 0 oder größer 344, ist der Gegner unsichtbar. Im anderen Falle muß das Sprite an der berechneten Position eingeschaltet werden. Diese Rechnung übernimmt das Unterprogramm »SPRITE« von Zeile 2930 bis 3490:

Die Bildschirmposition wird von der Spriteposition subtrahiert. Das Low-Byte des Ergebnisses gelangt ins y-Register, das High-Byte ins x-Register. Hat das High-Byte den Wert »0«, wird das Unterprogramm »SPRITEON« ausgeführt. Ist das Ergebnis weder »0« noch »1«, wird zu »SPRITEOFF« verzweigt.

Die Bewegung der Sprites geschieht im Unterprogramm »ENEMIES« ab Zeile 4130. Als Variable dient die Speicherstelle 821. Sie wird zunächst auf »0« gesetzt. Der Inhalt der Bewegungsrichtung des ersten Gegners »ENEMY1DIR« wandert in den Akku. Ist er »0« (Bewegungsrichtung links), wird von der 16-Bit x-Position 1 subtrahiert, ansonsten 1 addiert (Bewegungsrichtung rechts). Sollte beim Subtrahieren die x-Position = 160 sein, wird die Richtung durch Erhöhen von »ENEMY1DIR« auf »1« nach rechts geändert.

Im Unterprogramm »ENEMIES2« (Zeile 4590 bis 4860) geschieht ähnliches für die y-Position der Sprites. Der wichtigste Unterschied ist, daß wir die Werte 96 bis 152 als Bewegungsgrenzen festlegen.

Türen öffnen und schließen sich

Wir haben beschlossen, dafür Zeichenmuster im Zeichensatz zu ändern. In unserem Demo-Listing sind dies die Zeichen 68 bis 85. Wir wollen jeweils die Zeichen 68 bis 76 und die Zeichen 77 bis 85 durchscrollen lassen. Das heißt: 68 wird 76, 69 wird 68, 70 wird 69 bis zum Zeichen 76, es bekommt das Muster von 75. Das gleiche machen wir mit den Zeichen 77 bis 85. Dazu speichern wir zuerst die Zeichenmuster von 76 und 85 (je acht Byte). Dann scrollen wir alle Zeichenmuster ab 76 bis 85 um acht Byte (ein Zeichen). Zum Schluß übertragen wir die Muster der gespeicherten Zeichen in 68 und 77. Damit dies nicht zu schnell geschieht, bauen wir noch einen Verzögerungszähler ein. Im Listing reservieren wir zum Zwischenspeichern 16 Byte (von Zeile 8000 bis Zeile 8150) und für den Verzögerungszähler ein Byte (Zeile 8151). Ab Zeile 8152 beginnt das eigentliche Programm.

Wenn Sie ein eigenes Playfield entwickeln, sollten Sie berücksichtigen, daß die Zeichen 68 bis 85 später im Spiel ständig getauscht werden.

Animation von Raumschiff und Gegnern

Zur Animation des Raumschiffs haben wir acht Spriteformen in Spriteblock 32 ($0800) bis 39 zur Verfügung. Wir haben mit einem Kreis begonnen, der über immer kleiner werdende Elypsen bis zum Strich in Block 39 wird. Eine (scheinbare) Drehung erreichen wir durch zwei Phasen. Bei jedem Aufruf der Routine wird von Spriteblock 32 über 33 bis 39 umgeschaltet, dann von 39 über 38 bis 32. Beim Wechsel der Drehrichtung ändern wir die Spritefarbe, durch Invertieren von Bit 2 im Farbregister von Sprite »0«. Da sich unser Sprite zu schnell drehen würde, wenn wir bei jedem Rasterinterrupt aufrufen, verwenden wir auch hier eine Zählvariable.

Zeile 320 bis 590 sind für diese Aufgabe zuständig.

In dem Demo-Game sind allen Gegnern dieselben Spritemuster zugeteilt. Das vereinfacht die Konstruktion. Die Konstruktion eines sich drehenden Rings ist von Blocknummer 40 bis 47 abgelegt. Animiert wird analog der Routine für das Raumschiff. Im Listing sind dafür die Zeilen 6280 bis 6620 zuständig.

Kollision führt zum Spielende

Wir wollen das Spiel beenden, wenn unser Raumschiff mit dem Hintergrund oder einem Gegner zusammengestoßen ist. Dies ist wieder durch einen Interrupt des VIC möglich. Dieser Baustein ist in der Lage, durch eine Berührung zweier Sprites und/oder eines Sprite mit dem Hintergrund, einen IRQ auszulösen. Dazu ist Register 53274 (V+26) zuständig. Bit 1 erlaubt einen IRQ für Sprite/Hintergrund, Bit 2 den für Sprite/Sprite. In unserem Demo-Spiel machen wir von diesem Interrupt keinen Gebrauch. Für uns ist es uninteressant, wenn die Gegner mit dem Hintergrund kollidieren, daher müssen wir auf jeden Fall das Sprite/Hintergrund-Kollisionsregister (53279 = V+31) abfragen, wer mit wem zusammengestoßen ist. Jedes gesetzte Bit in diesem Register entspricht einer Sprite/Hintergrund-Berührung des entsprechenden Sprites. Beim zweiten Kollisions-Register (53278 = Sprite/Sprite) prüfen wir lediglich, ob überhaupt ein Bit gesetzt ist, da unsere Gegner nicht miteinander kollidieren können. Wenn wir dies ganz normal im Raster-Interrupt machen, sind wir im ungünstigsten Fall um 1/25 Sekunde später dran, als mit einem eigenen IRQ. Das stört uns nicht, wichtig ist, daß überhaupt reagiert wird. Im Demo-Programm erledigen diese Abfrage die Zeilen 2790 bis 2816 für uns. Die Maske in Zeile 2791 überprüft, ob das Raumschiff (Sprite 0 = Bit 0 = %00000001 = 1) bei der Kollision beteiligt war. Wenn ja, wird eine Spielvariable erhöht, und beim nächsten Rasterinterrupt das Spiel abgebrochen.

Initialisierung und Aufruf der Routinen

Das Wichtigste für unser Spiel fehlt uns noch: Die Initialisierung unserer Routinen und ihre Verbindung. Diesen Programmteil kann man nicht als eigenständiges Modul betrachten, da er als einziger für jede neue Spielvariante auch neu geschrieben werden muß. Überlegen wir uns, was wir alles initialisieren:

1. Blocknummer für Sprite 0 zuweisen, das Sprite positionieren und einschalten (schaltet die anderen aus).
2. Softscroll-Zeiger = »7«.
3. Spielstandsvariable = »0«
4. Sonstige Pointer und Variablen auf Ursprungswerte.
5. Bildschirm löschen, mit dem ersten Ausschnitt der Spiele-Landschaft füllen.
6. Bildschirm- und Rahmenfarben setzen.
7. Multicolor einschalten
8. Kollisionsregister löschen
9. Musikroutine vorbereiten
10. IRQ-Vektoren auf eigene Routine.

Im Programm geschieht dies durch die Zeilen 120 bis 140 (ab S. 32), wobei jeweils Unterroutinen aufgerufen werden.

Beispielsweise wird »INITIAL«, von Zeile 600 bis 860 nach jedem Spielende aufgerufen, wogegen IRQ (ab Zeile 880 nur zu Spielbeginn benötigt wird.

Der Spielablauf selbst gestaltet sich, dank der modularen Programmierung, noch einfacher:

Es ist die Routine »IRQNEU« von Zeile 2460 bis 2920 und ruft je nach Zustand der Spielevariablen entweder »WARTEN« (0), »BOING« (2) oder »OVER« (3) auf:

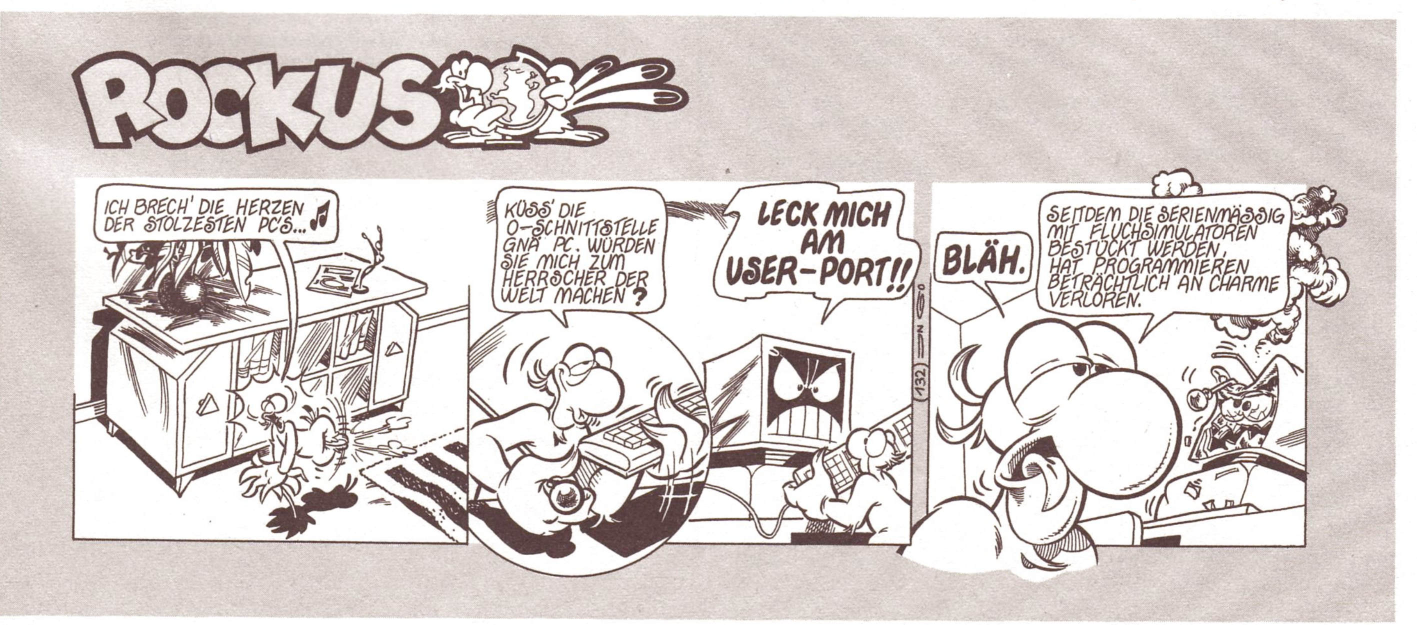
»WARTEN« ist die erste Routine, gibt den Text »Bitte Knopf drücken« (Text 1) aus und setzt nach »Feuer« die Spielvariable (Game, Zeile 1840) auf »1«.

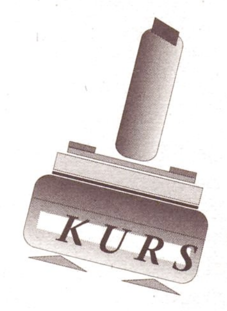
»BOING« tritt nach einem verpatzten Spiel in Kraft, gibt sinngemäß den Text »daneben« aus und löscht ihn nach einer Weile wieder. Danach wird »INITIAL« aufgerufen, die Spielevariable auf »0« gesetzt und beim nächsten IRQ zwangsläufig »WARTEN« aufgerufen.

»OVER« gibt »gewonnen« aus, wartet eine Weile, löscht danach den Text wieder und ruft »INITIAL« auf. Danach steht die Spielevariable wieder auf »0« und beim nächsten IRQ geht's wieder zu »WARTEN«.

SPIEL Ist die Spielevariable jedoch »1«, ist das Spiel in vollem Gange und die eigentlichen Spieleroutinen »SCROLL«, »ANIMATION« und »JOYAB« finden Verwendung.

Das Demo-Spiel dient nur zur Anregung eigener, vielleicht besserer Routinen. Es lassen sich z.B. in der Funktion »WARTEN« Highscores anzeigen, oder in »OVER« der Highscore in eine Liste eintragen. »OVER« könnte auch den IRQ ausschalten, eine Kennvariable belegen und in Zeile 140 würde dann nicht RTS stehen, sondern wenn diese Variable gesetzt ist, würde der nächste Level nachgeladen, der IRQ wieder auf die eigene Routine verbogen und der neue Level neu initialisiert. Durch die modulare Programmierung steht einer Erweiterung des Spiels nichts mehr im Wege. Wenn Sie jetzt unser





Kurzinfo: DEMO

Programmart: DEMO-File zum Spielekurs
Spielziel: Manövrieren
Sie ohne Karambolage durchs Labyrinth
Laden: LOAD"DEMOGAME",8
Starten: nach dem Laden RUN eingeben, danach SYS 12288
Steuerung: Joystick in Port 2
Besonderheiten: gesamtes Spiel läuft im IRQ
Benötigte Blocks: 27 (gepackt)
Programmautoren: Andreas von Lepel/N. Heusler

kleines Demo ausprobieren wollen, sollten Sie den Computer kurz ausschalten und es danach nochmal laden:
LOAD"DEMOGAME",8

Jetzt entpacken Sie mit RUN und starten mit SYS12288 (Abb. 7). Warum trotz des Spielablaufs ein Cursor am Bildschirm blinkt, ist Ihnen sicherlich auch klar. (gr)

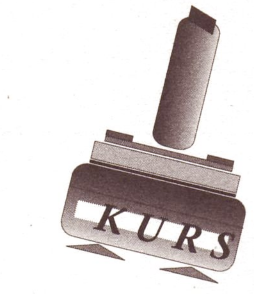
[7] Das komplette Spiel läuft im IRQ ab. Daher blinkt auch der Cursor am Bildschirm.



Das kommentierte Listing zum Kurs

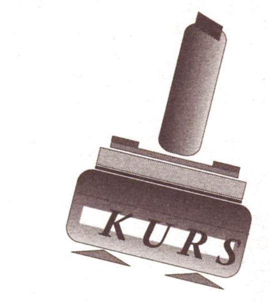
```
110 V =53248
120 JSR INITIAL ; ADRESSEN INIT
125 JSR INITSPR
130 JSR IRQ     ; IRQ UMLENKEN
140 RTS
145 ;
150 SETSPRITE
151 LDA #130    ; SPRITE 0 AUF
160 STA V       ; POSITION (130/114)
170 LDA #114
180 STA V+1
190 LDA #0
200 STA V+16
210 STA V+29
220 STA V+23
225 LDA #2      ; ROT
230 STA V+39    ; SETZEN
240 LDA #5      ; GRUEN
250 STA V+37    ; LISTCOLOR1
260 LDA #32     ; SPRITEPOINTER
270 STA 2040
280 LDA #1      ; LISTCOLOR AN
290 STA V+28
300 STA V+21    ; SPRITE AN
310 RTS
315 ;
320 ANIMATION:LDA ZEIT
330 CMP #5
340 BEQ DOANIMA
350 INC ZEIT
360 RTS
370 DOANIMA:LDA #0
380 STA ZEIT
390 LDA DREHEN
400 BNE ZU
405 ;
410 AUF:LDA 2040
420 CMP #32
430 BNE AUF2
440 INC DREHEN
450 JMP ZU2
460 AUF2:DEC 2040
470 RTS
475 ;
480 ZU:LDA 2040
490 CMP #39
500 BNE ZU2
510 DEC DREHEN
520 LDA V+39
530 EOR #4
540 STA V+39
550 JMP AUF2
560 ZU2:INC 2040
570 RTS
580 DREHEN:BRK
590 ZEIT:BRK
595 ;
600 INITIAL:LDA #39 ; SPR.0 AUS BL 39
610 STA 2040       ;
620 LDA #0         ; NULL
650 STA 65        ; LB GRAFIKPOINT = 0
660 STA GAME      ; SPIELSTAND     = 0
670 LDA #7        ; SOFTSCROLLZEIGER
680 STA 2         ; AUF SIEBEN
690 LDA #200      ; VERZ. NACH SPIELENDE
700 STA DELAY     ; 200/25 SEK.
710 LDA #$70      ; HB GRAFIKPOINT =$70
720 STA 66        ; ($7000 = GRAFIK)
730 LDA #<1144    ; BEGINN BILDSCHIRM
740 STA 67        ;
750 LDA #>1144    ; 67/68 = 1144
760 STA 68
770 LDA #5        ;
772 STA 53281     ; FUER ALTE 64'ER
780 STA 646       ; UND CURSORFARBE
790 JSR $E544     ; SCNSLR1
791 ;
795 JSR FARBINI
796 ;
800 LDA 53270     ; 38 ZEICHEN PRO ZEILE
810 AND #247      ; %11110111
820 STA 53270
830 JSR RAMSCREEN ; GRAFIK -> SCHIRM
840 JSR SETSPRITE ; SPRITE 0 AN
842 LDA #0        ; SCHWARZ
844 STA 53280     ; -> RAHMEN
846 STA 53281     ; -> HINTERGRUND
850 LDA 53279     ; LOESCHEN KOLL SPR/HGR
855 LDA 53278     ;          KOLL SPR/SPR
860 RTS
861 ;
870 DELAY:BRK   ; VERZOEGERUNGSZAEHLER
880 IRQ:SEI     ; IRQ AUS
881 ;
885 ; ********************************
890 ; ** AN DIESE STELLE KOMMT DIE **
900 ; **      INITIALISIERUNG      **
910 ; **  FUER DIE MUSIK IM IRQ    **
920 ; ********************************
955 ;
960 LDA #<IRQNEU   ; IRQ LB AUF EIGENE
970 STA $314       ; ROUTINE
980 LDA #>IRQNEU   ; IRQ HB AUF EIGENE
990 STA $315       ; ROUTINE
1000 LDA #74     ; RASTERZEILE 74
1010 STA V+$12   ; INS RASTERREG.
1020 LDA V+$11   ; ZEILE < 255
1030 AND #$7F    ; %01111111 (S.TABELLE)
1040 STA V+$11   ; DAHER 9.BIT AUS
1050 LDA #$81    ; RASTERZEILENIRQ VIC
1060 STA V+$1A   ; ZULASSEN
1070 CLI         ; IRQ WIEDER AN
1080 RTS
1090 GAME:BRK   ; SPIELSTANDVARIABLE
1095 ;
1100 JOYAB:LDA #224 ; TASTATUR
1110 STA 56322   ; UMSCHALTEN
1120 LDA 56320   ; JOYPORT 2
1130 AND #1      ;= BIT 1 GESETZT (UP)
1140 BNE JOY2    ; JA, DANN WEITER
```

```
1150 DEC V+1     ; SPRITE Y-POS -1
1160 JOY2:LDA 56320
1170 AND #2      ;= BIT 2 GESETZT (DOWN)
1180 BNE JOY3    ; JA, DANN WEITER
1190 INC V+1     ; SPRITE Y-POS +1
1200 JOY3:LDA 56320
1210 AND #4      ;= BIT 3 GESETZT (LEFT)
1220 BNE JOY4    ; JA, WEITER
1230 LDA V       ; WENN X-POS = 32
1240 CMP #32     ; IGNORIEREN
1250 BEQ JOY4
1260 DEC V       ; SONST X-POS -1
1270 JOY4:LDA 56320
1280 AND #8      ;= BIT 4 GESETZT (RIGHT)
1290 BNE JOY5    ; JA, WEITER
1300 LDA V       ; WENN X-POS = 255
1310 CMP #255    ;
1320 BEQ JOY5    ; IGNORIEREN
1330 INC V       ; SONST X-POS +1
1340 JOY5:LDA #255 ;
1350 STA 56322   ; TASTATUR WIEDER EIN
1360 RTS
1365 ;
1370 RAMSCREEN:LDX #0     ; 13 ZEILEN
1380 RAMSCRL1:LDY #0      ; 40 ZEICHEN
1390 RAMSCRL2:LDA (65),Y  ; AUS
1400 STA (67),Y           ; GRAFIKBEREICH
1410 INY                  ; ZUM
1420 CPY #40              ; BILDSCHIRM
1430 BNE RAMSCRL2         ; KOPIEREN
1440 INC 66
1450 INC 66       ; GRAFIKVEKTOR IST
1460 LDA 67
1470 CLC          ; IN DER ZERONEW
1480 ADC #40
1490 STA 67       ; 65/66
1500 LDA 68       ; BILDSCHIRMVEKTOR IST
1510 ADC #0
1520 STA 68       ; 67/68
1530 INX
1540 CPX #13       ; SCHON 13 ZEILEN
1550 BNE RAMSCRL1 ; NEIN
1560 LDA 66        ; GRAFIKVEKTOR
1570 SEC
1580 SBC #26
1590 STA 66        ; UND
1600 LDA #<1144    ; BILDSCHIRMVEKTOR
1610 STA 67
1620 LDA #>1144
1630 STA 68        ; RESTAURIEREN
1640 RTS
1645 ;
1650 WARTEN:LDY #0 ; TEXT1 AUSGEBEN
1660 WARTEN1:LDA TEXT1,Y
1670 STA 1754,Y
1680 INY              ; AB BSCH POS 1754
1690 CPY #20          ; 20 ZEICHEN
1700 BNE WARTEN1
1710 LDA #224   ; TASTATUR UMSCHALTEN
1720 STA 56322  ;
1730 LDA 56320  ; = JOYKNOPF GEDR.
1740 CMP #111   ; %01101111
1750 BNE WARTENEND ; NEIN, ENDE
1760 LDA #255   ; TASTATUR
1770 STA 56322  ; EIN
1780 LDY #0     ; TEXT1 LOESCHEN
1790 LDA #32    ; = SPACES DRUEBER
1800 WARTEN2:STA 1754,Y
1810 INY
1820 CPY #20
1830 BNE WARTEN2
1840 INC GAME   ; SPIELSTAND AUF 1
1850 WARTENEND:RTS
1860 TEXT1 .8,9,20,32,20,18,9
1862 .7,7,5,18,32,20
1865 .15,32,19,20,1,18,20
1866 ;
1870 BOING:LDY #0 ; TEXT2 AUSGEBEN
1880 BO1:LDA TEXT2,Y
1890 STA 1754,Y
1900 INY
1910 CPY #20     ; 20 ZEICHEN
1920 BNE BO1
1930 DEC DELAY  ; WARTEN BIS
1940 LDA DELAY  ; VERZOEGERUNG
1950 BNE BOEND  ; BEI NULL
1960 LDY #0
1970 LDA #32     ; TEXT2 LOESCHEN
1980 BO2 STA 1754,Y
1990 INY
2000 CPY #20
2010 BNE BO2
2020 JSR INITIAL ; ALLES NEU INITIEREN
2025 JSR INITSPR
2030 BOEND:RTS
2040 TEXT2 .25,15,21,18,32,13,9
2041 .19,19,9,15,14,32
2042 .6,1,9,12,5,4,32
2043 ;
2050 OVER:LDY #0  ; TEXT3 AUSGEBEN
2060 OVERL1:LDA TEXT3,Y
2070 STA 1754,Y
2080 INY
2090 CPY #20        ; 20 ZEICHEN
2100 BNE OVERL1
2110 DEC DELAY      ; WARTEN BIS
2120 LDA DELAY      ; VERZOEGERUNG
2130 BNE OVEREND    ; NULL IST
2140 LDY #0
2150 LDA #32        ; TEXT LOESCHEN
2160 OVERL2:STA 1754,Y
2170 INY
2180 CPY #20
2190 BNE OVERL2
2200 JSR INITIAL    ; ALLES INITIALISIEREN (NEUSTART)
2205 JSR INITSPR
2210 OVEREND:RTS
2220 TEXT3 .25,15,21,32,13,1,4,5
2221 .32,9,20,44,15,12
2222 .4,32,2,15,25,44
2223 ;
2230 SCROLL:LDA 2  ; 7 PUNKTE LINKS
2240 BEQ BLKSCROLL ; HARDSCROLLING
2250 DEC 2        ; EINEN PUNKT NACH LINKS
2260 LDA 65       ; ABFRAGEN
2270 CMP #$D8     ; OB DIE GRAFIK AM ENDE
2280 BNE BLKEND
2290 LDA 66       ; ANGELANGT IST
2300 CMP #$71     ; (BEI $71D8)
2310 BNE BLKEND   ; NEIN, WEITER
2320 INC GAME     ; GAME = GAME +2
2330 INC GAME
2340 BLKEND:RTS
2350 BLKSCROLL:LDA #7; VARIABLE SOFTSCR
2360 STA 2        ; AUF SIEBEN
2370 LDA 65       ; GRAFIKPOINTER +1
2380 CLC
2390 ADC #1
2400 STA 65
2410 LDA 66       ; HIGHBYTE
2420 ADC #0
2430 STA 66
2440 JSR RAMSCREEN ; TRANSFERROUTINE
2450 RTS
2455 ;
2460 IRQNEU:LDA V+$19 ; IRQ-REGISTER
2470 STA V+$19        ; LOESCHEN
2480 BMI RASTER ; WAR RASTER-IRQ!
2490 LDA $DC0D  ; SONST NORMALER IRQ
2492 ;
2493 ; **********************
2494 ; **   HIER AUFRUF    **
2495 ; ** AUFRUF DER MUSIK **
2496 ; **********************
2497 ;
2500 CLI         ; IRQ ZULASSEN
2510 JMP $EA31   ; ZUR SYSTEMROUTINE
2515 ;
2520 RASTER:LDA V+$12 ; RASTERIRQ
2530 CMP #178          ; BEI ZEILE 178
2540 BCS ZWEITER       ; JA, WEITER
2550 LDA 53270   ; AB ZEILE 74
2560 AND #248    ; UM DEN WERT IN
2570 ORA 2       ; SPEICHERZELLE 2
2580 STA 53270   ; NACH LINKS SCROLLEN
2590 LDA #5      ; RAHMENFARBE
2600 STA 53280   ; GRUEN
2601 LDA #12
2602 STA 53282
2605 JSR ENEMIES
2606 JSR ENEANI
2609 JSR CHARANI
2610 LDA #178    ; NEXT  IRQ ZEILE 178
2620 STA $D012
2630 JMP $FEBC   ; IRQ ENDE
2635 ;
2640 ZWEITER:LDA 53270 ; BILDSCHIRM AB
2650 AND #248          ; ZEILE 178
```




```
2660 ORA #7              ; NORMAL SCHALTEN
2670 STA 53270
2680 LDA #0      ; SCHWARZ
2690 STA 53280   ; RAHMENFARBE
2700 LDA GAME    ; SPIELSTAND
2710 BNE GAME1   ; NICHT NULL = WEITER
2720 JSR WARTEN ; (SPIELBEGINN)
2730 JMP EXIT
2740 GAME1:CMP #1
2750 BNE GAME2   ; NICHT 1 = WEITER
2760 JSR SCROLL     ; SCROLLEN (SPIEL)
2770 JSR ANIMATION ; SPRITE
2780 JSR JOYAB      ; JOYST. ABFRAGEN
2785 JSR ZAEHLEN
2790 LDA 53279    ; SPR/HGR KOLLISION
2791 AND #1
2800 BEQ NOBOING1; KEIN ZUSAMMENSTOSS
2810 INC GAME     ; SONST VERMERKEN
2811 JMP EXIT
2812 NOBOING1
2813 LDA 53278
2814 AND #1
2815 BEQ NOBOING
2816 INC GAME
2820 NOBOING:JMP EXIT
2825 ;
2830 GAME2:CMP #2
2840 BNE GAME3   ; NICHT 2 = WEITER
2850 JSR BOING   ; (KOLLISION)
2860 JMP EXIT
2865 ;
2870 GAME3 CMP #3
2880 BNE EXIT     ; NICHT 3, DANN ENDE
2890 JSR OVER     ; (SPIELENDE)
2900 EXIT:LDA #74 ; NAECHSTER IRQ
2910 STA V+$12     ; AB ZEILE 74
2920 JMP $FEBC    ; IRQ BEENDEN
2925 ;
2930 SPRITE:SEC    ;POSITIONSBERECHNUNG
2940 LDA POSENELO ; UND EIN/AUS
2950 SBC POSSCRLO
2960 TAY
2970 LDA POSENEHI
2980 SBC POSSCRHI
2990 TAX
3000 BEQ SPRITEON
3010 CMP #1
3020 BNE SPRITESAVE
3030 TYA
3040 CMP #89
3050 BCS SPRITESAVE
3060 LDA V+16
3070 ORA 820
3080 STA V+16
3090 JSR NUMMER
3100 LDA V+21
3110 ORA 820
3120 STA V+21
3130 LDA #1
3140 STA 821
3150 RTS
3155 ;
3160 SPRITESAVE
3161 LDX 820
3170 TXA
3180 EOR #255
3190 STA 820
3200 LDA V+16
3210 AND 820
3220 STX 820
3230 STA V+16
3240 LDX 820
3250 TXA
3260 EOR #255
3270 STA 820
3280 LDA V+21
3290 AND 820
3300 STX 820
3310 STA V+21
3320 LDA #0
3330 STA 821
3340 RTS
3345 ;
3350 SPRITEON:LDX 820
3360 TXA
3370 EOR #255
3380 STA 820
3390 LDA V+16
3400 AND 820
3410 STX 820
3420 STA V+16
3430 JSR NUMMER
3440 LDA V+21
3450 ORA 820
3460 STA V+21
3470 LDA #1
3480 STA 821
3490 RTS
3495 ;
3500 NUMMER:LDA 820
3510 CMP #2
3520 BNE NUMMER0
3530 STY V+2
3540 NUMMER0:CMP #4
3550 BNE NUMMER1
3560 STY V+4
3570 NUMMER1:CMP #8
3580 BNE NUMMER2
3590 STY V+6
3600 NUMMER2:CMP #16
3610 BNE NUMMER3
3620 STY V+8
3630 NUMMER3:CMP #32
3640 BNE NUMMER4
3650 STY V+10
3660 NUMMER4:CMP #64
3670 BNE NUMMER5
3680 STY V+12
3690 NUMMER5:CMP #128
3700 BNE NUMMER6
3710 STY V+14
3720 NUMMER6:RTS
3725 ;
3730 ENEMY1XLO:BRK ;DATEN FUER SPR-
3740 ENEMY1XHI:BRK ;POSITION
3750 ENEMY1Y   :BRK
3760 ENEMY1DIR:BRK
3770 ENEMY2XLO:BRK
3780 ENEMY2XHI:BRK
3790 ENEMY2Y   :BRK
3800 ENEMY2DIR:BRK
3810 ENEMY3XLO:BRK
3820 ENEMY3XHI:BRK
3830 ENEMY3Y   :BRK
3840 ENEMY3DIR:BRK
3850 ENEMY4XLO:BRK
3860 ENEMY4XHI:BRK
3870 ENEMY4Y   :BRK
3880 ENEMY4DIR:BRK
3890 ENEMY5XLO:BRK
3900 ENEMY5XHI:BRK
3910 ENEMY5Y   :BRK
3920 ENEMY5DIR:BRK
3930 ENEMY6XLO:BRK
3940 ENEMY6XHI:BRK
3950 ENEMY6Y   :BRK
3960 ENEMY6DIR:BRK
3970 ENEMY7XLO:BRK
3980 ENEMY7XHI:BRK
3990 ENEMY7Y   :BRK
4000 ENEMY7DIR:BRK
4010 POSSCRLO :BRK
4020 POSSCRHI :BRK
4030 POSENELO :BRK
4040 POSENEHI :BRK
4045 ;
4050 ZAEHLEN:LDA POSSCRLO
4060 CLC
4070 ADC #1
4080 STA POSSCRLO
4090 LDA POSSCRHI
4100 ADC #0
4110 STA POSSCRHI
4120 RTS
4130 ENEMIES:LDA #0
4140 STA 821
4150 LDA ENEMY1DIR
4160 BEQ LINKS1
4170 LDA ENEMY1XLO
4180 CLC
4190 ADC #1
4200 STA ENEMY1XLO
4210 LDA ENEMY1XHI
4220 ADC #0
4230 STA ENEMY1XHI
4240 LDA ENEMY1XLO
4250 CMP #70
4260 BNE VIEW1
```

```
4270 LDA ENEMY1XHI
4280 CMP #1
4290 BNE VIEW1
4300 DEC ENEMY1DIR
4310 JMP VIEW1
4320 LINKS1:LDA ENEMY1XLO
4330 SEC
4340 SBC #1
4350 STA ENEMY1XLO
4360 LDA ENEMY1XHI
4370 SBC #0
4380 STA ENEMY1XHI
4390 LDA ENEMY1XLO
4400 CMP #160
4410 BNE VIEW1
4420 LDA ENEMY1XHI
4430 CMP #0
4440 BNE VIEW1
4450 INC ENEMY1DIR
4460 VIEW1:LDA ENEMY1XLO
4470 STA POSENELO
4480 LDA ENEMY1XHI
4490 STA POSENEHI
4500 LDA #2
4510 STA 820
4520 JSR SPRITE
4530 LDA 821
4540 BEQ ENEMIES2
4550 LDA ENEMY1Y
4560 STA V+3
4570 LDA #1
4580 STA V+40
4590 ENEMIES2:LDA #0
4600 STA 821
4610 LDA ENEMY2DIR
4620 BEQ OBEN1
4630 INC ENEMY2Y
4640 LDA ENEMY2Y
4650 CMP #152
4660 BNE VIEW2
4670 DEC ENEMY2DIR
4680 JMP VIEW2
4690 OBEN1:DEC ENEMY2Y
4700 LDA ENEMY2Y
4710 CMP #96
4720 BNE VIEW2
4730 INC ENEMY2DIR
4740 VIEW2:LDA ENEMY2XLO
4750 STA POSENELO
4760 LDA ENEMY2XHI
4770 STA POSENEHI
4780 LDA #4
4790 STA 820
4800 JSR SPRITE
4810 LDA 821
4820 BEQ ENEMIES3
4830 LDA ENEMY2Y
4840 STA V+5
4850 LDA #1
4860 STA V+41
4870 ENEMIES3:LDA #0
4880 STA 821
4890 LDA ENEMY3DIR
4900 BEQ OBEN2
4910 INC ENEMY3Y
4920 LDA ENEMY3Y
4930 CMP #152
4940 BNE VIEW3
4950 DEC ENEMY3DIR
4960 JMP VIEW3
4970 OBEN2:DEC ENEMY3Y
4980 LDA ENEMY3Y
4990 CMP #96
5000 BNE VIEW3
5010 INC ENEMY3DIR
5020 VIEW3:LDA ENEMY3XLO
5030 STA POSENELO
5040 LDA ENEMY3XHI
5050 STA POSENEHI
5060 LDA #8
5070 STA 820
5080 JSR SPRITE
5090 LDA 821
5100 BEQ ENEMIES4
5110 LDA ENEMY3Y
5120 STA V+7
5130 LDA #1
5140 STA V+42
5150 ENEMIES4:LDA #0
5160 STA 821
5170 LDA ENEMY4DIR
5180 BEQ OBEN3
5190 INC ENEMY4Y
5200 LDA ENEMY4Y
5210 CMP #152
5220 BNE VIEW4
5230 DEC ENEMY4DIR
5240 JMP VIEW4
5250 OBEN3:DEC ENEMY4Y
5260 LDA ENEMY4Y
5270 CMP #96
5280 BNE VIEW4
5290 INC ENEMY4DIR
5300 VIEW4:LDA ENEMY4XLO
5310 STA POSENELO
5320 LDA ENEMY4XHI
5330 STA POSENEHI
5340 LDA #16
5350 STA 820
5360 JSR SPRITE
5370 LDA 821
5380 BEQ ENEMIES5
5390 LDA ENEMY4Y
5400 STA V+9
5410 LDA #1
5420 STA V+43
5430 ENEMIES5:LDA #0
5440 STA 821
5450 LDA ENEMY5DIR
5460 BEQ OBEN4
5470 INC ENEMY5Y
5480 LDA ENEMY5Y
5490 CMP #152
5500 BNE VIEW5
5510 DEC ENEMY5DIR
5520 JMP VIEW5
5530 OBEN4:DEC ENEMY5Y
5540 LDA ENEMY5Y
5550 CMP #96
5560 BNE VIEW5
5570 INC ENEMY5DIR
5580 VIEW5:LDA ENEMY5XLO
5590 STA POSENELO
5600 LDA ENEMY5XHI
5610 STA POSENEHI
5620 LDA #32
5630 STA 820
5640 JSR SPRITE
5650 LDA 821
5660 BEQ ENEMIES6
5670 LDA ENEMY5Y
5680 STA V+11
5690 LDA #1
5700 STA V+44
5710 ENEMIES6:LDA #0
5720 STA 821
5730 LDA ENEMY6DIR
5740 BEQ OBEN5
5750 INC ENEMY6Y
5760 LDA ENEMY6Y
5770 CMP #152
5780 BNE VIEW6
5790 DEC ENEMY6DIR
5800 JMP VIEW6
5810 OBEN5:DEC ENEMY6Y
5820 LDA ENEMY6Y
5830 CMP #96
5840 BNE VIEW6
5850 INC ENEMY6DIR
5860 VIEW6:LDA ENEMY6XLO
5870 STA POSENELO
5880 LDA ENEMY6XHI
5890 STA POSENEHI
5900 LDA #64
5910 STA 820
5920 JSR SPRITE
5930 LDA 821
5940 BEQ ENEMIES7
5950 LDA ENEMY6Y
5960 STA V+13
5970 LDA #1
5980 STA V+45
5990 ENEMIES7:LDA #0
6000 STA 821
6010 LDA ENEMY7DIR
6020 BEQ OBEN6
6030 INC ENEMY7Y
6040 LDA ENEMY7Y
6050 CMP #152
6060 BNE VIEW7
```




```
6070 DEC ENEMY7DIR
6080 JMP VIEW7
6090 OBEN6:DEC ENEMY7Y
6100 LDA ENEMY7Y
6110 CMP #96
6120 BNE VIEW7
6130 INC ENEMY7DIR
6140 VIEW7:LDA ENEMY7XLO
6150 STA POSENELO
6160 LDA ENEMY7XHI
6170 STA POSENEHI
6180 LDA #128
6190 STA 820
6200 JSR SPRITE
6210 LDA 821
6220 BEQ ENEMIES8
6230 LDA ENEMY7Y
6240 STA V+15
6250 LDA #1
6260 STA V+46
6270 ENEMIES8:RTS
6275 ;
6280 ENEANI:LDA ZEIT2 ;VERZOEGERUNG
6290 CMP #2          ;TESTEN (=2)
6300 BEQ DOANIMA2
6310 INC ZEIT2
6320 RTS
6330 DOANIMA2:LDA #0
6340 STA ZEIT2
6350 LDA DREHEN2     ;SPRITE ANIMATION
6360 BNE ZU3
6370 AUF3:LDA 2041
6380 CMP #40
6390 BNE AUF4
6400 INC DREHEN2
6410 JMP ZU4
6420 AUF4:DEC 2041   ;BLOCKPOINTER
6430 DEC 2042        ;ERNIEDRIGEN
6440 DEC 2043
6450 DEC 2044
6460 DEC 2045
6470 DEC 2046
6480 DEC 2047
6490 RTS
6495 ;
6500 ZU3:LDA 2041    ;2. PHASE
6510 CMP #47
6520 BNE ZU4
6530 DEC DREHEN2
6540 JMP AUF4
6550 ZU4:INC 2041    ;BLOCKPOINTER
6560 INC 2042        ;ERHOEHEN
6570 INC 2043
6580 INC 2044
6590 INC 2045
6600 INC 2046
6610 INC 2047
6620 RTS
6630 DREHEN2:BRK
6640 ZEIT2:  BRK
6645 ;
6650 INITSPR:LDA #0  ;BSCHPOSITION
6660 STA POSSCRLO    ;AUF NULL
6670 STA POSSCRHI
6680 LDA #43         ;POINTER FUER
6690 STA 2041        ;ENEMIES
6700 STA 2042        ;FESTLEGEN
6710 STA 2043
6720 STA 2044
6730 STA 2045
6740 STA 2046
6750 STA 2047
6760 LDA #16        ;POSITIONEN
6770 STA ENEMY1XLO  ;ENEMIES
6780 LDA #1
6790 STA ENEMY1XHI
6800 LDA #126
6810 STA ENEMY1Y
6820 LDA #12
6830 STA ENEMY2XLO
6840 LDA #3
6850 STA ENEMY2XHI
6860 LDA #100
6870 STA ENEMY2Y
6880 LDA #52
6890 STA ENEMY3XLO
6900 LDA #3
6910 STA ENEMY3XHI
6920 LDA #121
6930 STA ENEMY3Y
6940 LDA #92
6950 STA ENEMY4XLO
6960 LDA #3
6970 STA ENEMY4XHI
6980 LDA #142
6990 STA ENEMY4Y
7000 LDA #132
7010 STA ENEMY5XLO
7020 LDA #3
7030 STA ENEMY5XHI
7040 LDA #100
7050 STA ENEMY5Y
7060 LDA #172
7070 STA ENEMY6XLO
7080 LDA #3
7090 STA ENEMY6XHI
7100 LDA #121
7110 STA ENEMY6Y
7120 LDA #212
7130 STA ENEMY7XLO
7140 LDA #3
7150 STA ENEMY7XHI
7160 LDA #142
7170 STA ENEMY7Y
7180 RTS
7185 ;
8000 CHSAVE76:BRK   ;FREIHALTER FUER
8010 BRK            ;ZEICHENSATZ-
8020 BRK            ;ANIMATION
8030 BRK            ;16 BYTE = 2ZEICHEN
8040 BRK
8050 BRK
8060 BRK
8070 BRK
8080 CHSAVE85:BRK
8090 BRK
8100 BRK
8110 BRK
8120 BRK
8130 BRK
8140 BRK
8150 BRK
8151 CHARDELAY:BRK
8152 CHARANI:LDY #0  ;ZEICHENSATZ-
8161 INC CHARDELAY   ;ANIMATION
8162 LDA CHARDELAY
8163 CMP #5
8164 BEQ CHARGOON1
8165 RTS
8166 CHARGOON1:STY CHARDELAY
8167 LDA #0
8170 CHLOOP1:LDA 8800,Y
8180 STA CHSAVE76,Y
8190 LDA 8872,Y
8200 STA CHSAVE85,Y
8210 INY
8220 CPY #8
8230 BNE CHLOOP1
8240 LDY #135
8250 LDX #143
8260 CHLOOP2:LDA 8736,Y
8270 STA 8736,X
8280 DEX
8290 DEY
8300 CPY #255
8310 BNE CHLOOP2
8320 LDY #0
8330 CHLOOP3:LDA CHSAVE76,Y
8340 STA 8736,Y
8350 LDA CHSAVE85,Y
8360 STA 8808,Y
8372 INY
8380 CPY #9
8390 BNE CHLOOP3
8400 RTS
8500 FARBINI:LDA #12
8510 STA 53282
8520 LDA #11
8530 STA 53283
8540 LDA 53270
8550 ORA #16
8560 STA 53270
8565 LDA #24
8566 STA 53272
8570 LDY #0
8580 LDA #9
8590 FARBLOOP:STA 55296,Y
8600 STA 55546,Y
8610 STA 55796,Y
8620 STA 56041,Y
8630 INY
8640 BNE FARBLOOP
8650 RTS
8652 ENDE
```

W. P. Tennis II – Grand Slam 2100

Spiel, Satz und Sieg

Wer spielt mindestens so gut Tennis wie Edberg, Becker und Courier? Die Roboter in unserer futuristischen Sportspiel-Simulation für einen oder zwei Spieler!

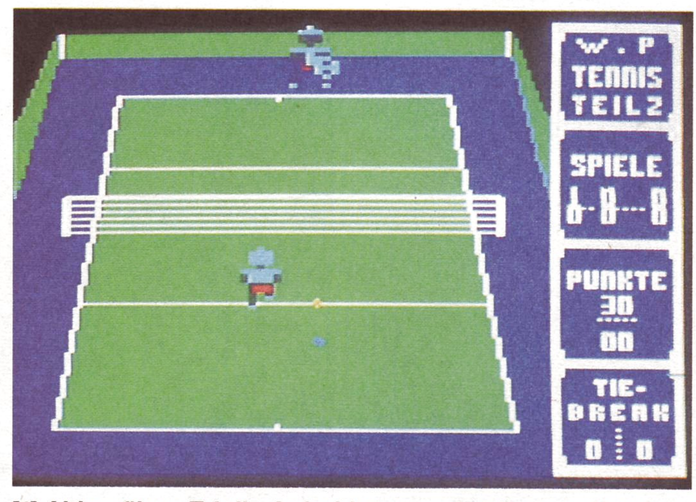
Im Jahr 2100 hat sich Tennis zum Volkssport Nr. 1 gemausert. Die Menschheit ist aber inzwischen noch bequemer geworden, daher läßt man spielen: Zwei Roboter treten gegeneinander an. Auch die Regeln haben sich seit den beschaulichen Tagen in Wimbledon geändert: Man schlägt jetzt in der Feldmitte auf, der Aufschlag wechselt nach jedem Punkt (nicht erst nach einem Spiel!). Da sich das herkömmliche Punktesystem bewährt hatte, wurde es beibehalten. Die blechernen Spieler bestreiten zwei Gewinnsätze. Steht's 6 : 6, entscheidet der Tie-Break.

Laden Sie das Spiel mit:

```
LOAD "W.P TENNIS II",8
```

und geben Sie RUN ein. Es dauert nun einige Sekunden, bis das Programm entpackt ist (normalerweise würde es 95 Blöcke auf der Diskette belegen und unsere Sonderheft-Diskette aus den Angeln heben!). Nun müssen Sie erneut RUN eingeben, um das Spiel zu starten.

Zunächst werden Ihnen per Menü diverse Einstellmöglichkeiten geboten (Abb. 2). Achtung! Es müssen unbedingt beide Joysticks in den entsprechenden Ports stecken, auch wenn Sie alleine gegen den Computer spielen möchten! Sie brauchen dann zwar nur noch Joystick 1, zum Einstellen benützt man aber Joystick 2!



[1] Altbewährte Taktik: Aufschlagen und sofort ans Netz!

Kurzinfo: W.P. Tennis II

Programmart: Tennissimulation
Laden: LOAD "W.P TENNIS II",8
Starten: nach dem Laden RUN eingeben
Steuerung: Joystick Port 1 und 2
Besonderheiten: für einen oder zwei Spieler, verschiedene Spielstärken sind einstellbar.
Benötigte Blocks: 27 (nach dem Entpacken: 95)
Programmautor: Wolfgang Polenda

Schnelles Spiel: Joystick 2 nach oben,
langsam: Joystick 2 nach unten,
Ein-Spieler-Modus: Joystick 2 nach rechts,
Zwei-Spieler-Modus: Joystick 2 nach links drücken.

Beachten Sie, daß man im Ein-Spieler-Modus gegen den Computer spielt. Als einsamer Fighter müssen Sie unbedingt den Joystick in Port 1 benutzen: Dann kontrollieren Sie den Tennis-Roboter im unteren Spielfeld.

Hat man die Modi eingestellt, kommt man per Feuerknopf in den nächsten Menübildschirm:

Serve & Volley: Joystick 2 nach oben,
Allround-Spieler 1: Joystick 2 nach unten,
Allround-Spieler 2: Joystick 2 nach rechts,
Grundlinienspieler: Joystick 2 links.

Nach der entsprechenden Wahl bewegen Sie den Joystick nach rechts unten und drücken gleichzeitig den Feuerknopf: Der Tennis-Court erscheint. Die beiden Roboter-Cracks stehen an der Grundlinie. Am rechten Bildschirmrand erkennt man die Anzeigetafel: Spiele, Punkte und Tie-Break.

Das Match beginnt durch Druck auf den Feuerknopf in Port 1 (oder die Leertaste). Mit dem Joystick 1 steuert man die Spielfigur im unteren Feld, mit Joystick 2 den Tennis-Roboter in der oberen Spielhälfte. Sie können die Spieler in alle Richtungen bewegen. Ist der Ball nahe genug, muß man die entsprechende Schlaghand per <Joystick links oder rechts> aktivieren und den Feuerknopf drücken. In dieser Sekunde läßt sich die Spielfigur allerdings nicht mehr bewegen: Falls Sie sich verrechnet haben, saust der Ball vorbei und kann nicht mehr zurückgeschmettert werden – wieder 15 Minuszähler!

Ist das Match vorbei (einer der Spieler muß zwei Sätze gewonnen haben), erscheint erneut das Einstellmenü (wie zu Programmbeginn). Jetzt lassen sich andere Modi einstellen und ausprobieren.

```
WENN SIE DEN 1.SPIELER MODUS GE-
WAEHLT HABEN(ALSO EIN SPIEL GEGEN
DEN COMPUTER),STELLEN SIE JETZT
BITTE DIE STAERKE DES COMPUTERS
EIN UND DRUECKEN SIE DANN DEN
FEUERKNOPF + JOYSTICK NACH UNTEN
RECHTS ZUM START.
JOYSTICK NACH OBEN    = STAERKE 1
JOYSTICK NACH UNTEN   = STAERKE 2
JOYSTICK NACH RECHTS  = STAERKE 3
JOYSTICK NACH LINKS   = STAERKE 4
 COPYRECHTE BY WOLFGANG POLENDA
```

[2] Interessante Spielmodi bietet das Einstellmenü

Spieltaktik

Wer allein gegen den Computer spielt, wird dessen Taktik schnell erkennen: Der Gegner schlägt auf und geht dann direkt ans Netz (Abb. 1). So erwischt er fast jeden Ball, den Sie retournieren. Unser Rat: Machen Sie's nach! Dann werden Sie den gegnerischen Roboter sehr schnell in Schwierigkeiten bringen. Unschlagbar ist er nicht – sonst wäre die Tennis-Simulation der Zukunft schon in der Gegenwart langweilig! Eines werden Sie rasch feststellen: Die Maschinenmännchen können mit den bekannten Tennis-Cracks der Gegenwart leicht mithalten.

(Wolfgang Polenda/bl)

Flowers – Blumen brauchen Wasser

... WENN NELKEN WELKEN

Als Gärtner mit Herz und Gewissen müssen Sie außerirdische Riesenblumen ausreichend mit frischem Wasser versorgen.

Blumen sind hochsensibel und verlangen die berühmte »Grüne Hand«. Besonders, wenn es sich um die seltene Art Riesennelken vom Planeten »Flower Power« handelt, die Sie für 1000 intergalaktische Dollar auf dem letztjährigen Flohmarkt in Saturn-City erworben haben. Die seltenen Pflänzchen müssen ständig mit Wasser versorgt werden, sonst lassen Sie schnell den Köpfe hängen!

Stellen Sie zunächst Ihre Gießkanne in die Ecke, laden Sie das Spiel mit:

```
LOAD "FLOWERS",8
```

und geben Sie RUN ein. Nach dem Entpacken muß man erneut RUN eintippen und die RETURN-Taste drücken: Das Spiel startet.

Auf dem Bildschirm erscheint Level 1. Sie stecken nun den Joystick in Port 2 und drücken die Leertaste – das Gewächshaus mit vier Riesennelken erscheint (Abb. 1). Der winzige Sprite am unteren Bildschirm ist unser Mini-Gärtner, der sich nun schleunigst die Gießkanne schnappen sollte: Die Blumen machen schon bedenklich schlapp! Dabei möchte unser Hobby-Gärtner an der »Großen Universum-Blumen-Show« teilnehmen, die nächsten Sonntag stattfindet. Klammheimlich sieht er sich nämlich schon als Preisträger (Platz 1, wenn's geht!).

Per Joystick muß man das Männchen möglichst schnell zum Stengel der gefährdeten Blume bringen, damit es kräftig gießen kann. Die Wirkung ist offensichtlich: Sofort bewegt sich die Blüte wieder nach oben. Außerdem erhöht sich die Score-Anzeige links unten. Die anderen Blumenköpfe aber setzen ihre Talfahrt fort – man muß sie ständig im Auge behalten und hin- und herflitzen, um überall hilfreich einzugreifen.

[1] Diese vier Pflänzchen machen mehr Arbeit als eine ganze Plantage

Ein altes Gärtner-Sprichwort heißt: »Allzu viel ist ungesund!«. Wenn Sie einer Nelke zuviel Aufmerksamkeit schenken, wächst Sie durchs Glasdach. Die herabfallenden Glassplitter zerstören dann alle Weltmeisterschaftsträume.

Als Spieler sollte man den goldenen Mittelweg wählen: Verteilen Sie das kostbare Wasser gezielt – ein wenig hier, ein bißchen dort. Jeder, der sich bereits in der Schule mit mathematischen Knobeleien herumgeärgert hat, kann sich ausrechnen, daß bei noch so intensiver Joystickakrobatik irgendwann der Zeitpunkt kommt, an dem alle Blüten auf den Boden klatschen (Abb. 2). Dann hilft kein Rennen und kein Gießen: Die Spielrunde ist vorbei! (Abb. 3)

Allerdings besitzt der Gärtner eine wirksame Medizin, die er gezielt einsetzen muß: Fünf Reset-Pillen, die man mit <RUN/STOP> bei der entsprechenden Blume aktiviert. Dann macht die Blüte einen Riesensprung nach oben. Dadurch kann man sich um die anderen »Schlaffis« kümmern.

(Lewin Eisele/bl)



[2] 100 Eimer Wasser geschleppt – alle verwelkt!

Kurzinfo: Flowers

Programmart: Geschicklichkeit
Laden: LOAD "FLOWERS",8
Starten: nach dem Laden RUN eingeben
Steuerung: Joystick Port 2
Besonderheiten: mit <RUN/STOP> können pro Level fünf Reset-Pillen aktiviert werden
Benötigte Blocks: 7
Programmautor: Lewin Eisele



[3] Das Spiel ist zwar vorbei, aber der High score präsentiert sich recht mager und mickrig

Hyperthrust - rasende Luftballons im Weltraum

Kautschuk-Olympiade

Machen Sie mit beim irrwitzigen Wettlauf zweier Gummibälle über die zerklüftete Oberfläche von »Gumball«. Achten Sie dabei aufs Zeitlimit!

Gehören Sie zur Meisterklasse der »Letzten Ninjas vom Großen Joystick«? Dann begleiten Sie uns zum fernen Himmelskörper »Gumball«, dessen Oberfläche aus federndem, elastischem Gummi besteht. Wen wundert's, daß die Bewohner keine menschliche Gestalt haben, sondern runde Gummibälle sind!

Eines haben sie mit den Menschen gemeinsam: unbekümmerten Spieltrieb und Freude am sportlichen Wettkampf. Deshalb finden jedes Jahr die G.O.G (Gum Olympic Games) statt, an denen sich junge, sportliche Bälle des Gummi-Planeten beteiligen.

Mit Spannung wartet das Publikum auf den Höhepunkt der Sportspiele, den 5000-Meilen-Lauf quer übers ganze Land. Um den Wettkampf abwechslungsreicher zu gestalten, wurde ein Zeitlimit von 4000 Zehntelsekunden gesetzt. Wer aber weiß, wie sportlich durchtrainiert diese Gummibälle sind, dem scheint die Zeit nicht zu knapp. Außerdem werden gegnerische Kautschukteile ins Feld geschickt, die als Hindernisse für die Läufer fungieren. Jeder Sportler besitzt eine Strahlenpistole, die man mit dem Feuerknopf aktiviert. Damit müssen die Hindernisse geschmolzen werden (außerdem gibt's dafür eine Menge Punkte).

Geteilter Bildschirm

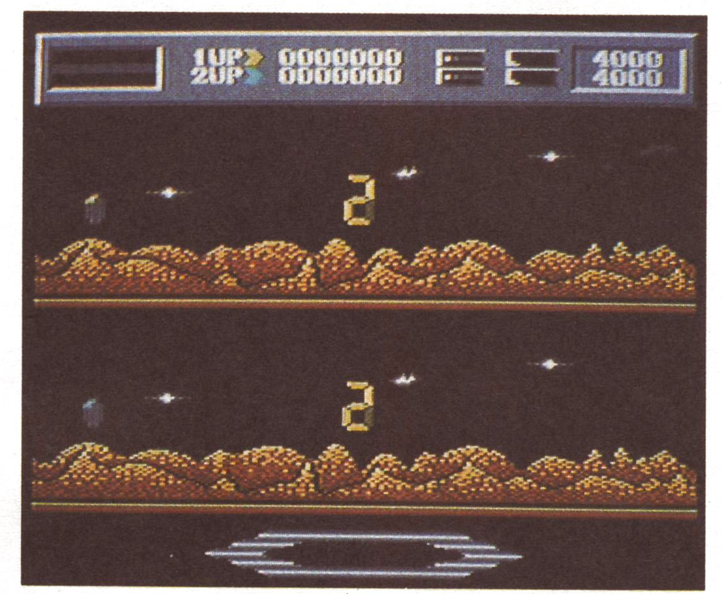
Die beiden Wettkämpfer stehen erwartungsvoll am Start: Sie wissen genau, daß sie von Ihnen nach besten Kräften mit dem Joystick unterstützt werden!

Da bis zum Beginn des Wettrennens nicht mehr viel Zeit bleibt, laden Sie das Spiel mit:

```
LOAD "HYPERTHRUST",8
```

und starten mit RUN.

Der Startbildschirm erscheint. Mit der Leertaste oder dem Feuerknopf (Joystick Port 2) kommt man weiter. Wichtig sind die drei Möglichkeiten, das Spiel ablaufen zu lassen (Abb. 1).



[1] Die Spannung steigt, der Gummi hüpft

Wählen Sie die gewünschte Spielart, indem Sie den Joystick auf- oder abwärts bewegen:

- Shootem Double (zwei Spieler und zwei Joysticks),
- Shootem Solo (ein Spieler, nur Joystick Port 2),
- Double Tactics (ebenfalls für zwei Spieler).

Durch erneuten Start per Feuerknopf geht's zum Start (Abb. 2). Noch fünf Sekunden - dann beginnt die wilde Hatz!

Der Bildschirm ist in zwei Bahnen geteilt. Am oberen Rand erkennt man die Score-Anzeigen für beide Spieler (Nr. 1 benutzt die obere Bahn und den Joystick 2, Nr. 2 die untere Rennstrecke mit dem Joystick in Port 1: Bitte nicht verwechseln!). Daneben steht die Zeitanzeige, die bei »4000« beginnt.

Sie müssen sich unwahrscheinlich sputen, um die Strecke in dieser kurzen Zeit zurückzulegen (Joystick kräftig nach rechts drücken, und ballern, was das Zeug hält!). Denn mindestens so wichtig wie das Zeitlimit sind die Punkte, die man unterwegs sammelt. Dicke Punkte- und Zeit-Boni bekommen Sie für jeden Gummibrocken mit dem »B«.

Um sich im High score zu verewigen, genügt es, mehr Punkte zu erreichen, als der 10. Rang auf dem Konto ausweist. Vorher müssen Sie noch Ihren Namen eintragen. Die neue Bestenliste wird auf Diskette zurückgespeichert.

»Hyperthrust« benutzt Direktzugriffsblöcke. Achtung: Diese Diskettenseite darf nicht validiert werden! Außerdem läßt sich das Spiel nur mit einem Diskettenkopierprogramm 1:1 übertragen.

Wir wünschen Ihren Gummiklößen viel Erfolg beim sportlichen Wettkampf! (bl)

[2] Sorry, in dieser Runde haben Sie sich zu lange Zeit gelassen!



Kurzinfo: Hyperthrust

Programmart: Action-Game
Laden: LOAD "HYPERTHRUST",8
Starten: nach dem Laden RUN eingeben
Steuerung: Joystick Port 1 und 2
Besonderheiten: für ein oder zwei Spieler, High score wird auf Diskette gespeichert
Benötigte Blocks: 327
Programmautoren: U. Hafner/M. Bläske

Omnibus GmbH – Einsteigen und Türen schließen!

NICHTS ALS BUSSE

Würden Sie gerne Geschäftsführer bei der »Omnibus GmbH« sein? Unsere Handelssimulation stellt Ihre Managerfähigkeiten auf eine harte Probe! Man geht schneller Pleite, als man denkt...

Tagein, tagaus – immer dieselbe Strecke, zur Arbeit oder zur Schule. Ganz automatisch. Stutzig wird man erst, wenn der Bus ausbleibt – und mit zehnminütiger Verspätung eintrudelt: Dann ist man sauer und schimpft auf den MVV (Münchner Verkehrsverbund), die Berliner Verkehrsbetriebe (BVB) oder den Verkehrsverbund Rhein-Ruhr (VRR) – oder wen's sonst trifft. Haben Sie schon einmal darüber nachgedacht, wie viele Rädchen ineinandergreifen müssen, damit so ein Nahverkehrsnetz reibungslos funktioniert?

Bei »Omnibus GmbH« sind Sie Oberbefehlshaber einer Bus-Armada. Sie sollen aber nicht nur möglichst viel Bares verdienen, sondern vor allem ein gutes Image bei Ihren Fahrgästen aufbauen. Das wird von vielen Faktoren beeinflußt: Zustand der Fahrkartenverkaufsstellen und Fahrzeuge, Werbung und Ihr Verhältnis zum Finanzamt.

Steigen Sie ein ins Management mit:

```
LOAD "OMNIBUS GMBH",8
```

Das Spiel startet mit RUN. Legen Sie den Joystick getrost zur Seite: Zur Steuerung der Programmfunktionen reicht die Tastatur aus. Wenn das Titelbild (Abb. 1) erscheint, kommt man durch zweimaliges Drücken der Leertaste ins Spiel. Jetzt können Sie eine früher abgebrochene Version laden (J). Als Beispiel finden Sie »OB.VIER« auf unserer Diskette zum Sonderheft. An dieser Runde nahmen vier Spieler teil. Bei entsprechender Aufforderung dürfen Sie nur das Wort »VIER« eingeben (der Zusatz OB wird vom Programm automatisch angefügt).

Für ein neues Spiel drückt man <N>. Geben Sie nun die Anzahl der Mitspieler (1 bis 4) und deren Namen an. Jetzt lädt der Computer das Hauptprogramm (OMNI.MAIN). Zu Spielbeginn sieht es nicht allzu rosig aus: Sie besitzen einen Kurzbus (ein »Pullman« ist bereits bestellt und bezahlt); eine Linie ist ebenfalls schon in Betrieb (von Iserlohn nach Hemer). Am Kleingeld hapert's gewaltig: Mit 10000 Mark und 50 Aktien einer unbedeutenden Firma kreist der Pleitegeier schon bedrohlich über Ihrem jungen Unternehmen.

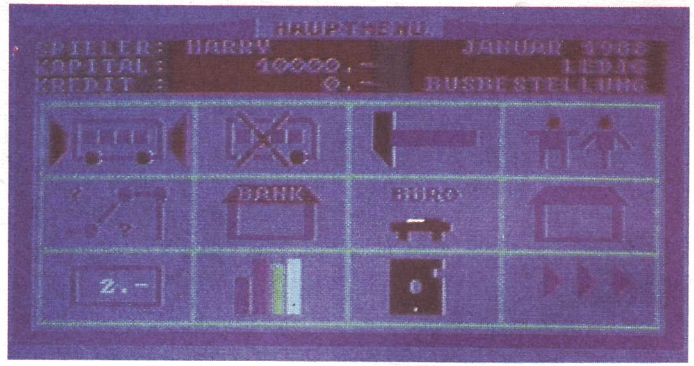


[1] Das Titelbild wurde mit dem Malprogramm »Koala-Painter« entworfen

Das Spielmenü (Abb. 2) wird sichtbar. In der oberen Menüleiste findet man Spielernamen, Geschäftsmonat, Anfangskapital, Kreditsumme und die aktuelle Menüfunktion. Das entsprechende Logo wird von zwei roten Balken umrahmt, deren Position sich mit den Cursortasten (links, rechts, aufwärts, abwärts) verändern läßt. Zur Geschäftsführung stehen Ihnen zwölf Auswahlmöglichkeiten zur Verfügung, die man mit <RETURN> aktiviert. In den folgenden Untermenüs dient ein violetter Pfeil zur Auswahl der Funktionen (<CRSR aufwärts/abwärts>):

Busbestellung:
Zu Spielbeginn besitzt man einen Kurzbus. Wenn's Ihre Finanzen zulassen, können Sie weitere Fahrzeuge kaufen. Die Ausgaben erhöhen die Kreditanzeige, daher muß man stets darauf achten, genügend Kreditwürdigkeit nachzuweisen. Lange Lieferfristen für die neuen Busse sind aber unvermeidbar.

Busverkauf:
Notfalls sollten Sie Busse verkaufen, wenn Ihnen die Schulden über den Kopf wachsen. Mit der Plustaste kann man den Verkaufspreis erhöhen. Der Käufer drückt dann die Taste, die seiner Preisvorstellung entspricht. Dazu muß er aber ausreichend Bargeld besitzen. Hat der Käufer den Preis über Gebühr gedrückt, kann es vorkommen, daß sich ein Konkurrenzunternehmen einschaltet und ihm den Bus vor der Nase wegschnappt. Also – nicht zu hoch pokern! Mit <RETURN> bricht man die Verkaufsaktion ab.



[2] Alle geschäftlichen Transaktionen wählt man im Hauptmenü

Kurzinfo: Omnibus GmbH

Programmart: Handelssimulation
Laden: LOAD "OMNIBUS GMBH",8
Starten: nach dem Laden RUN eingeben
Steuerung: Tastatur
Besonderheiten: Am Spiel können sich mindestens ein und maximal vier Spieler beteiligen. High-score-Liste auf Diskette wird aktualisiert.
Benötigte Blocks: 189
Programmautor: Christian Frantzke

Reparaturen:
Oberstes Gebot bei der Personenbeförderung ist die Verkehrssicherheit der Busse. Leider kommen sie nach jeder Fahrt immer weniger im Top-Zustand zurück: Abgenutzte Sitze und Motorenverschleiß sind nicht zu vermeiden. Diese Mängel müssen schleunigst behoben werden, sonst vergraulen

Sie die Fahrgäste. Ist ein Bus total verkehrsuntüchtig, spielt der Computer TÜV: Das Fahrzeug darf nicht mehr auf die Straße und bringt Ihnen keinen Pfennig Umsatz! Der aktuelle Zustand wird in Prozentzahlen angegeben. Schieben Sie also notwenige Reparaturen nicht auf die lange Bank!

Personal:
Damit stellen Sie Ihre Belegschaft zusammen: Busfahrer, Kontolleure, Putzfrauen, Verkäufer und Techniker. Mit <CRSR aufwärts/abwärts> wählt man die Berufsgruppe, per <CRSR rechts> kommt man in die Gehaltsspalte. Personalstand und Löhne lassen sich mit der Plustaste erhöhen und durch die Minustaste reduzieren. Denken Sie dran: Gutbezahltes Personal ist arbeitswilliger. Oder: Wenn Sie zuwenig Kontrolleure einstellen, steigt die Zahl der Schwarzfahrer.

Als Gradmesser für den physischen Zustand der Bus-Kapitäne dient der Fahrerindex. Notfalls müssen Sie den einen oder anderen zum Lehrgang schicken, was aber ein tiefes Loch in die Kasse reißt.

Linien:
Damit bauen Sie eine Linienstrecke auf, die (hoffentlich) gewinnbringend arbeiten wird:

– Linienplanung: Teilen Sie die Nahverkehrstrecken ein (z.B. von Hagen nach Dortmund). Behalten Sie die Kosten im Auge: Das Personal muß ebenfalls verplant werden, d.h. für jede Strecke brauchen Sie Fahrer, Kontrolleure und Putzkolonnen. Bei der Busverteilung muß man darauf achten, daß möglichst gleich viele Busse auf den Strecken verkehren – auch wenn die Rendite der einen oder anderen Linie alles andere als verlockend ist!

Wählen Sie die gewünschte Strecke aus und ändern Sie mit < + > bzw. <-> die Bus- und Personalanzahl. Man kann auch völlig neue Strecken einrichten – aber nur, wenn Bargeld lacht. <RETURN> bringt Sie zurück ins Hauptmenü.

– Streckenhäuschen: Es lockt mehr Fahrgäste an, wenn sie nicht im Regen stehen müssen. Sorgen Sie dafür, daß eine ausreichende Menge Häuschen auf der Strecke verteilt sind.

[3] Ihre Busse fahren nur auf den hellen Linien: die Streckenkarte

– Streckenkarte: Sie bietet eine übersichtliche Grafik des Nahverkehrsnetzes (Abb. 3). Blaue Linien und Streckenhäuschen gehören (noch) nicht zu Ihrem Fahrplan. Helle Markierungen weisen bestehende Fahrtstrecken aus.

Bank:
Unabdingbar für jedes rentable Unternehmen: Der richtige Umgang mit Geld. Eventuelle Geschäftsverluste lassen sich durch trickreichen Handel mit Gold und Aktienspekulationen ausgleichen.

Büro:
Ist die Verwaltungsarbeit noch so lästig – ohne sie geht gar nichts im Geschäftsleben. Sehen Sie mindestens einmal täglich im Büro vorbei und erledigen Sie den Schreibkram. Sie können z.B. Werbefeldzüge in Rundfunk und Presse starten oder Gebäude ausbessern lassen. Alle Maßnahmen erhöhen die Punktzahl Ihres Image-Kontos. »Übersicht« bringt den aktuellen Stand Ihrer Managertätigkeit auf den Bildschirm. Monatlich einmal müssen Sie auch die Steuererklärung machen.

Verkäufer gesucht

Sie können sich entscheiden, ob Sie dem Finanzamt gegenüber »ehrlich«, »normal« oder »betrügerisch« agieren. Ob es sich im weiteren Spielverlauf auszahlt, ist eine andere Frage: Werden Sie beim Mogeln erwischt, ist's schlecht fürs Image!

Geschäfte:
Schwarzfahren ist ungesetzlich, doch muß der Fahrgast auch die Möglichkeit haben, bequem ans Ticket zu kommen. Je größer das Liniennetz, desto mehr Fahrkartenverkaufstellen müssen Sie besitzen. Allerdings: Nicht alle arbeiten rentabel. Wenn Sie eine wieder verkaufen möchten, müssen Sie sich mit Geldverlust abfinden.

Jede Verkaufsstelle braucht einen Verkäufer, den Sie im Menüpunkt »Personal« rekrutieren müssen. Bei zu geringer Verkäuferzahl bleiben alle unbesetzten Ticketschalter geschlossen.

Fahrpreise:
Für jede Linie muß man vernünftige Fahrpreise eintragen oder die bestehenden übernehmen. Sind sie zu teuer, werden Ihre Busse bald leer sein; wenn man zu billig ist, schaut kein Gewinn raus. Neue Fahrpreise können Sie aber nur vereinbaren, wenn die Strecke zu Ihrem Liniennetz gehört!

Spielstand:
zeigt dem frischgebackenen Unternehmer, wie fest er auf geschäftlichen Füßen steht: Kapital, Kredit, Stufen, Busse, Strecken und Anzahl der Fahrkartenverkaufsstellen.

Abspeichern:
Wem die Managerpflichten über den Kopf wachsen, kann den aktuellen Spielstand auf Diskette speichern und später weitermachen.

Weiter:
Damit lösen Sie den eigentlichen Spielablauf aus (Ereignis-Modus). Dazu muß man die Frage nach dem nächsten Spieler mit <J> beantworten (auch wenn Sie nur allein spielen). Jetzt präsentiert Ihnen das Programm in jeder Runde neben guten Nachrichten auch eine Fülle von Ereignissen, die sich nicht erfreulich aufs Firmengeschehen auswirken und erhebliche Verluste mit sich bringen:

Angebote:
Vom Zufallsprinzip bestimmt, macht Ihnen das Spiel ab und zu verlockende Angebote, z.B. einen supergünstigen Bus zu kaufen (nur gegen Bargeld!), einen lukrativen Werbevertrag abzuschließen oder ein verbilligtes Fahrkartengeschäft zu erwerben.

Steuer:
Wenn Sie bei der Steuerklärung gemogelt haben, können Sie leicht erwischt und zu einer saftigen Nachzahlung verdonnert werden. Außerdem reduziert sich das Image-Konto.

Lieferungen:
Hier erhalten Sie Nachricht, ob und wann die bestellten Busse endlich eintreffen.

Die anschließende Auswertung des letzten Monats zeigt Ihnen die Anzahl der Fahrgäste auf den einzelnen Strecken, die aktuellen Zinssätze, sowie die Gold- und Aktienpreise. Erneuter Tipp auf <RETURN> bringt die Aufstellung des Geschäftmonats mit Gewinn- und Verlustrechnung aller Strecken. Bewegen Sie dazu den reversen Cursorbalken auf und ab. »Sonstiges« gibt Ihnen Informationen zu den übrigen Gehältern und Einnahmen aus Werbeverträgen. <RETURN> aktiviert das Menü für Busbestellungen.

(Christian Frantzke/bl)

Detektiv 2000 – Gauner, Gangster & Co.

Countdown in Main-City

Manlow schaut auf die Armbanduhr. Deren Sekundenzeiger rückt unaufhörlich vorwärts: Nur noch eine Stunde Zeit! Helfen Sie dem »Detektiv 2000« bei der Klärung seines verzwickten Falls! Sally ist verschwunden ...

Bei dieser Zukunftsvision in Form eines Abenteuerspiels geht's im Jahr 2069 kriminell zu in Main-City, der US-Metropole:

Ganze Stadtteile sind in der Hand von »Outlaws«, gesetzlosem Gesindel, das sich auf Überfälle, Rauschgift-Dealereien und Einbrüche spezialisiert hat.

Die finstersten Viertel sind die Saturnring-Siedlungen: Arbeitslose Mutanten, defekte Roboter und Androiden aus allen Galaxis-Richtungen haben sich dort eingenistet. Die Staatspolizei ist machtlos, obwohl man mindestens einmal in der Woche eine Großrazzia durchzieht. Leider fängt man nur die kleinen Fische, an die großen Gangster kommt man nicht ran...

Schlüpfen Sie in die Rolle von Dwight Donald (abgekürzt: D. D.) Manlow, dem ausgefuchstesten Privat-Detektiv der tristen Stadt. Sally, die Tochter von J. J. Cloud, einem der reichsten Männer von Main-City, ist gekidnappt worden. D. D. Manlow hat den Auftrag, das Mädchen vor der Lösegeldübergabe wiederzufinden und dem Vater wohlbehalten ans Herz zu drücken. Es ist jetzt 23.00 Uhr, in einer Stunde läuft das Ultimatum ab. Manlow wirft die leere Whiskeyflasche in den Papierkorb, nimmt seinen Hut, die 38er und knallt die Tür seines schäbigen Büros hinter sich zu. Das Abenteuer kann beginnen...

Laden Sie das Spiel mit:

```
LOAD "DETEKTIV 2000",8,1
```

Das Programm startet automatisch und meldet sich mit dem Titelbild (Abb. 1). Mit der Leertaste geht's weiter.

Sie haben die Wahl zwischen einer Spielanleitung (Taste <J>) oder dem direkten Start (mit <N>).

Der erste Situationsbericht: Manlow steht auf der Straße vor seinem Auto. Ein Gewitter zieht auf, es beginnt bereits zu tröpfeln...

Am unteren Bildschirmrand sieht man, wie die Zeit unaufhörlich verrinnt. Das Spiel bietet Ihnen verschiedene Aktionsmöglichkeiten (Abb. 2), die Sie mit den entsprechenden Tasten aufrufen müssen. Ein Beispiel:

```
Du
(a) steigst ein
(b) ...walking in the rain
```

Drücken Sie nur die Tasten <A> oder <B>. Richtungsangaben sind nicht notwendig. Wenn unter der Zeitanzeige die Meldung »Weiter: w« erscheint, müssen Sie auf <W> tippen: Sie erhalten Hinweise zur bestehenden oder kommen in die nächste Spielsituation. »Sterben« können Sie nur, wenn Sie den Detektiv Manlow anders reagieren lassen, als es das Adventure vorsieht.

Allerdings: Es ist nicht so einfach, immer die richtige Entscheidung zu treffen. Was Ihnen logisch erscheint, muß noch lange nicht der korrekte Schritt auf dem Weg zur Lösung sein!



[1] Entschlossener Blick, trotzig vorgeschobenes Kinn: D. D. Manlow kennt keine Furcht

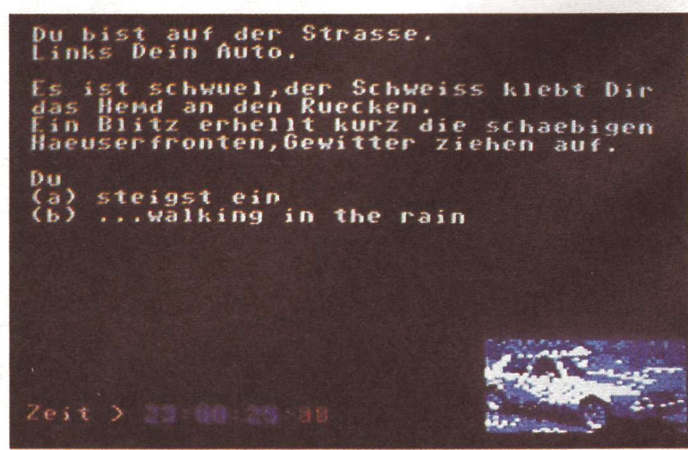
Sollten Sie ins Fettnäpfchen getreten sein, macht Sie das Spiel mit launischen Kommentaren darauf aufmerksam. Manlow kann viele Tode sterben, wenn er z.B.:

- bei der Explosion von 50 Kilo TNT in die Luft fliegt,
- sich im Spielecasino (Abb. 3) wie Rambo benimmt,
- vergiftete Cocktails schlürft,
- mit Betonschuhen an den Füßen im Abwasserkanal landet,
- in den Strahl einer Laserpistole gerät.

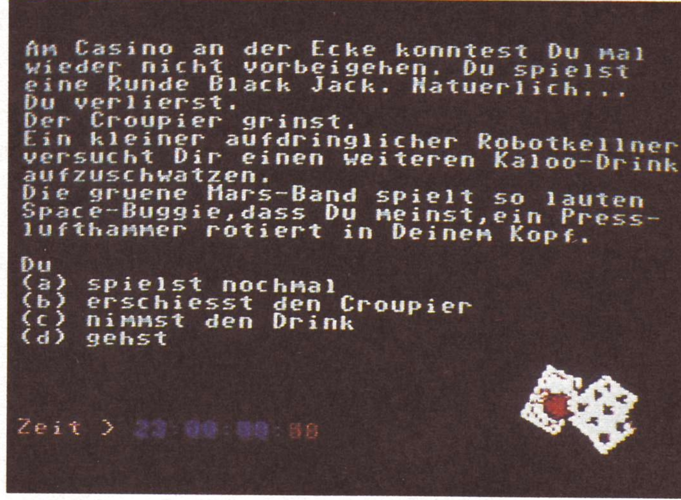
Genügt diese kleine Auswahl, Ihnen eine Gänsehaut über den Rücken zu jagen? Wenn man ein Adventure-Leben verloren hat, muß man das Spiel mit <S> neu starten. Fairerweise beginnt der Computer aber die Zeit wieder von Beginn an zu zählen (23.00 Uhr).

Videospiel und knifflige Rätsel inbegriffen

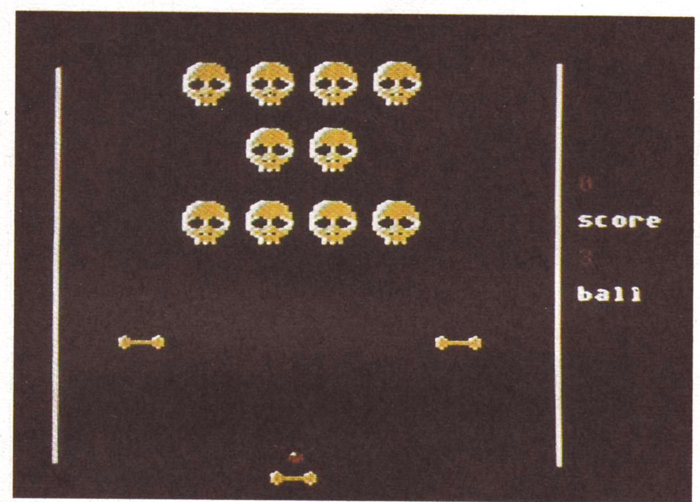
Aber es kommt noch besser: Geschickter Umgang mit dem Joystick ist in diesem Abenteuerspiel ebenso gefragt wie Grips und Kombinationsgabe. Ausgerechnet ein Plasma-Freß-Mutant, der hinter einigen schmutzigen Mülltonnen wohnt, verstellt Manlow den Weg. Er will ihn nur dann verschonen, wenn er mit ihm eine Runde »Doomball« über drei



[2] Nimmt er den Wagen oder geht er zu Fuß?



[3] Im Spiel-Casino: Manlow verfällt der Spielsucht und vergißt seinen Auftrag



[4] Zehn Totenschädel laden zum Tanz mit dem Joystick und gut geöltem Feuerknopf

Levels auf dem Video-Kommunikator spielt. Es bleibt dem Detektiv nichts anderes übrig, als den Joystick zu zücken, im Game-Port 2 anzuschließen und sein Bestes zu geben. Zehn aufgereihte Totenköpfe muß man mit einem Ball wegfegen; als Schläger dient ein Knochen, der mit dem Joystick in alle vier Richtungen bewegt werden kann (Abb. 4). Eines ist unabdingbar: Sie können im Adventure erst weitermachen, wenn Sie alle drei Levels dieses Zwischenspiels bewältigt haben. Denken Sie auch an die Zeit, die unaufhaltsam verrinnt...

Weitere Klippen sind die feinsinnigen Rätselfallen, die der Programmierer eingebaut hat und die den Spieler zur Verzweiflung treiben. Die zehn Fragen, die in verschiedenen Szenen des Adventures auftauchen, müssen richtig beantwortet werden (Lösungswort per Tastatur eintippen), sonst bleiben Sie an dieser Stelle hängen.

Daß die Quizfragen schwer sind, ist eine leichte Untertreibung: Ihr Knobelgeschick wird auf die Zerreißprobe gestellt! Oder wissen Sie auf Anhieb, wie man folgende Buchstabenreihe richtig ergänzt?

`R,R,R,R,T,I,I,I`

Falsch, weit daneben! Hier möchten wir Ihnen den ersten und einzigen Lösungstip zu diesem Adventure geben: Die Buchstaben haben nichts mit deren Position im Alphabet oder anderen geheimnisvollen Dingen zu tun. Es handelt sich jeweils um den letzten Buchstaben der zwölf Monate, von hinten aufgereiht:

- DEZEMBE (R),
- NOVEMBE (R),
- OKTOBE (R),
- SEPTEMBE (R),
- AUGUS (T),
- JUL (I),
- JUN (I),
- MA (I).

Die vier Buchstaben, auf die es ankommt, finden Sie nun sicher selbst heraus. Man muß sie, durch Kommas getrennt, eingeben und anschließend die RETURN-Taste drücken. Damit verhindern Sie, daß Manlow seine Lizenz als Privat-Schnüffler verliert.

Falls Sie schon vorher Gehirntraining betreiben möchten, hier die weiteren Rätselfragen (in der Reihenfolge, in der sie auftauchen):

Rätsel 2: Sie war erst eins, dann zwei, dann wieder eins.

Rätsel 3: Wie heißt die Lieblings-Oldie-Band des Programmautors? Die Kerle spielten 1990 den Song »Azzuro«.

Rätsel 4: Ausgeflippte Omega-Androiden reagieren auf lautes Pfeifen: Es deaktiviert ihre Angriffs-Sensoren. Um die Roboter endgültig abzuschalten, muß man den chiffrierten Code in Klartext eingeben. Die Chiffre-Zahlen:

`(2,16,32,24,2,40,38,18,10,8,30,6,36,10,8)/2`

Wie heißt der Code? Ein Tip: Es sind fünf Buchstaben. Betrachten Sie die Zahlenreihe als Rechenaufgabe. Überprüfen Sie die einzelnen Ergebnisse durch POKEn dieser Werte in den Bildschirmspeicher!

Rätsel 5: Was ist ca. 2,54 cm hoch, hat einen Durchmesser von 7,62 cm und wird mit Schlägern gespielt?

Rätsel 6: 100 Jahre ist es her, da bezwang er den König der Nacht. Wie heißt der Mann?

Rätsel 7: Zwei Formen muß der Karateka üben: Kata und...?

Rätsel 8: Was hat am morgen vier, mittags zwei und abends drei Beine?

Rätsel 9: An welchen Filmklassiker denken Sie bei »Play it again, Sam«?

Rätsel 10: Der venusianische Name des Gegenmittels bei einem Kreislaufkollaps ist »E-SO-CULC«. Übersetzen Sie das Wort ins Irdische und geben Sie die deutsche Übersetzung ein!

Daß der Programmautor ein langjähriger Adventure-Freak ist, erkennt man an humorigen Beschreibungen bestimmter Spielszenen, die eher in »Hobbit«, »Hitchhiker's Guide through the Galaxy« oder im Rollenspiel »Bard's Tale« angebracht wären. Wenigstens entschuldigt er sich wortreich für diese »Mißgriffe«.

Wenn Sie endlich am Ziel sind und Sally gefunden haben, wird Sie eine Information vom Hocker hauen, die den Fall in einem völlig anderen Licht erscheinen läßt. Die Frage ist nun, wie Sie sich entscheiden: Nur dann löst man das Adventure zu aller Zufriedenheit!

Das Spiel arbeitet mit der Echtzeitfunktion: Sie haben tatsächlich eine Stunde zur Verfügung, um den Lösungsweg bis zur letzten Spielszene zu finden. Achtung: Im Schlußbild kann man das Spiel nur per Reset verlassen.

Wir wünschen Ihnen eine schnupfenfreie Spürnase bei der Lösung dieses kniffligen Falls. Hüten Sie sich davor, das ganze Spiel tierisch ernst zu nehmen: Genau das wollte der Autor nicht! Bevor Sie sich vor Verzweiflung die Haare raufen: Den kompletten Lösungsweg erhalten Sie, wenn Sie uns einen frankierten Rückumschlag zusenden (Adresse bitte nicht vergessen!).

(H. W. Müller/bl)

Kurzinfo: Detektiv 2000

Programmart: Adventure
Laden: LOAD "DETEKTIV 2000",8,1
Starten: Spiel startet nach dem Laden automatisch
Steuerung: Tastatur und Joystick Port 2
Besonderheiten: Zu den meisten Spielszenen werden Grafiksequenzen eingeblendet. Integriertes Geschicklichkeitsspiel.
Benötigte Blocks: 154
Programmautor: H. W. Müller

Mic's Push'em
– verflixte Kistenschieberei

Großreinemachen

Roboter räumen ein Lagerhaus auf und verzetteln sich dabei total. Helfen Sie mit, alle grünen Kisten zu sortieren und an die richtige Stelle zu setzen!

Das Spiel ist eine überarbeitete Version von »Push'em« im 64'er-Sonderheft 49. Die 16 Levels wurden neu entworfen und ausgetauscht.

Sie brauchen viel Geschick, Übersicht und vor allem eine passende Strategie, damit Sie sich bei »Mic's Push'em« nicht den Weg verbauen.

Laden Sie das Spiel mit:

```
LOAD "MIC'S PUSH 'EM",8
```

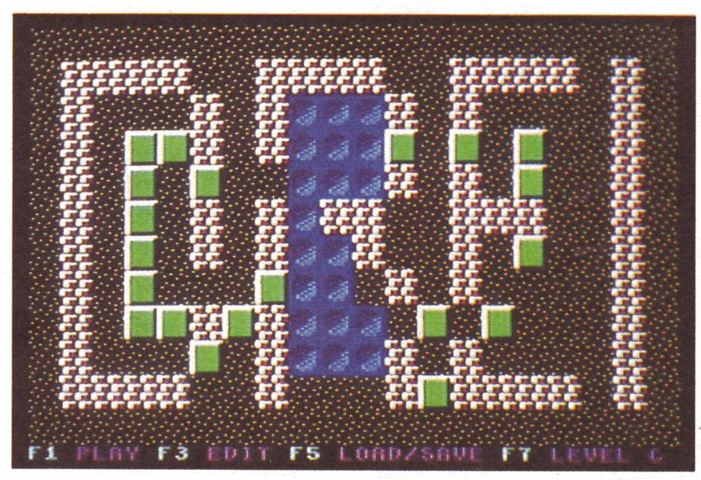
und starten es mit RUN.

Nach dem Entpacken meldet sich der Bildschirm mit dem ersten Level (Abb. 1). Durch die Funktionstasten <F3> und <F5> lassen sich zwei weitere Menüs aktivieren (Tabelle), <F7> schaltet den nächsten Level ein (A bis P). Das Spiel wird mit <F1> gestartet.

Als Roboter fungiert eine Silberkugel, die per Joystick in Port 2 gesteuert wird. Die Gänge in den Lagerhallen sind eng. Die grünen, quadratischen Kisten müssen in den blauen Mulden versenkt werden. Dazu plaziert man die Roboterkugel unmittelbar beim Karton, drückt den Feuerknopf und bewegt den Joystick in die gewünschte Richtung. Man kann die Kiste nur um jeweils eine Position weiterbewegen, ohne erneut den Joystick zu drücken. Stehen zwei oder mehr Kisten hintereinander, versagt die Kraft des Roboters: Er kann nur freistehende Kisten anstupsen. Befindet sich so ein Karton endlich auf der blauen Abstellfläche, färbt er sich ebenfalls blau. Wenn Sie sich verkalkuliert haben, lassen sich die Kisten von den Abstellflächen wieder abziehen und woanders plazieren. Der untere Spielfeldrand gibt Auskunft über die Anzahl der Roboterbewegungen (MOVES) und Kistentransporte (PUSHES).

Sehr schnell hat man sich in ausweglose Situationen manövriert, in denen es weder vorwärts noch rückwärts geht – dann hilft nur noch die Notbremse <F7>. Damit kehrt man in den ersten Level zurück.

Komplett gelöst ist ein Spielfeld, wenn sich alle grünen Kisten auf den entsprechenden Abstellplätzen befinden. Dann geht's ins nächste Level. Wenn Sie in einer höheren Spielstufe beginnen möchten, können Sie das zu Spielbeginn mit <F7> (ein Level vorwärts) und <F8> (rückwärts) bestimmen. <F1> startet die gewählte Spielstufe. Unmittelbar darauf erscheint die Spielgrafik.



[1] Die untere Menüleiste bietet zusätzliche Einstellmöglichkeiten

Das Spiel bietet durch den integrierten Editor die Möglichkeit, eigene Levels zu entwerfen. Mit <F3> aktivieren Sie diese Funktion. Am unteren Bildschirmrand erscheint eine geänderte Menüleiste:

<F1> Screen: Links oben wird ein quadratischer Cursorblock sichtbar, der sich ebenfalls mit dem Joystick bewegen läßt. Per Feuerknopf plaziert man ein Spielfeld-Element an der aktuellen Cursorposition.

Der Editor-Modus

<F3> Sprite: Damit schaltet die Joystickkontrolle vom Grafikcursor auf die Roboterfigur um. Sie läßt sich an gewünschter Position plazieren.

<F5> Item: Die Farbe des Grafikcursors ändert sich. Sie weist aufs aktuelle Spielfeld-Element hin:

- Gelb: Leerfeld,
- Hellblau: Abstellmulde,
- Grün: Kiste,
- Blau: bereits abgestellter Karton,
- Rot: Mauer.

<F7> Exit: bringt Sie zurück zum Spiel.

Das neu entworfene Spielfeld ersetzt während der restlichen Spieldauer den aktivierten Level.

Wer seine Kreationen dauerhaft auf Diskette verewigen und später wieder laden möchte, wählt im Hauptmenü die Funktion <F5> (LOAD/SAVE). Die neue Menüleiste bietet jetzt die Möglichkeiten, das Levelblid mit <F3> zu speichern oder per <F1> zu laden. Geben Sie dazu beim Strichcursor den entsprechenden Dateinamen ein. Achtung: Pro Spielrunde dürfen Sie lediglich 16 Levels verwenden!

(F. Thiesse/M. Reineke/bl)

Taste	Funktion	Wirkung
Hauptmenü		
<F1>	PLAY	Spielbeginn (Rücksprung mit <F7>)
<F3>	EDIT	Sprung in den Editor
<F5>	LOAD/SAVE	Sprung zum LOAD-/SAVE-Menü
<F7>/<F8>	LEVEL	Spielstufe anwählen
EDIT-Menü		
<F1>	SCREEN	mit Joystick Spielelement plazieren
<F3>	SPRITE	Startposition der Spielfigur festlegen
<F5>	ITEM	Spielelement stufenweise anwählen *Bedeutung der Cursorfarben:* Gelb = Leerfeld, Hellblau = Abstellplatz, Grün = Kiste, Blau = Kiste auf Abstellplatz, Rot = Mauer
<F7>	EXIT	zurück ins Hauptmenü
LOAD-/SAVE-Menü		
<F1>	LOAD	Spielfeld laden
<F3>	SAVE	Spielfeld speichern (Vor <F1> und <F3> mit Tastatur Filenamen eingeben)
<F7>	EXIT	zurück ins Hauptmenü

Alle Menüfunktionen auf einen Blick

Kurzinfo: Mic's Push'em

Programmart: Strategie-Geschicklichkeitsspiel
Laden: LOAD "MIC'S PUSH 'EM",8
Starten: nach dem Laden RUN eingeben
Steuerung: Joystick Port 2
Besonderheiten: Spiel besitzt integrierten Editor zum Entwerfen neuer Spielfelder
Benötigte Blocks: 64
Programmautoren: Frederic Thiesse/Michael Reineke

Tron – wie auf Schienen über Platinen

Tödliche Barrieren

Selten war ein wahnwitzigeres Rennen auf der Leinwand zu sehen, als im Sciene-fiction-Film »Tron«. Unser Spiel wiederholt (so oft Sie wollen) diese rasanten Szenen auf dem Bildschirm.

Die Computer-Animation in diesem Streifen stammt von den Lucas-Film-Studios in USA. »Tron« kann zwar nicht mit solch perfekter Grafik aufwarten, verfolgt aber dasselbe Prinzip: Zwei futuristische Motorräder, angetrieben durch Laser-Energie, jagen quasi in den Tiefen Ihres C64 über Platinen und Schaltungen (im Spiel durch ein Gitterraster kenntlich gemacht). Zur Erläuterung der Spielidee: Die beiden Gegner (im Film sind es zwei Computerprogramme) rasen übers Spielfeld und lassen auf ihrem Weg eine Energiebarriere zurück, die sich in der jeweiligen Fahrspur dahinter aufbaut. Stößt der Gegner gegen dieses Hindernis, ist's vorbei mit ihm. Achtung: Rennt man gegen die eigene Energiemauer, geht's einem nicht besser – Eigentor!

Laden Sie das Spiel mit:

```
LOAD "TRON",8
```

und starten Sie es mit RUN.

Nach dem Entpacken erscheint das Spielmenü (Abb. 1), in dem Sie die gewünschten Modi per Funktionstasten einstellen:

<F1>: Hier wählen Sie, wieviele Spieler (einer oder zwei) teilnehmen. Im »One-Player-Mode« wird der Gegner durch den C64 simuliert.

<F2>: Damit können Sie Sark, Ihren Gegner, ein- oder ausschalten. Dann jagen Sie allein durch die Tiefen der Bits und Bytes.

<F3>: Diese Funktion reagiert anders als die letztgenannte. Hier verschwindet Sark ebenfalls vom Spielfeld, hat aber nur eine Tarnkappe aufgesetzt: Er nimmt nach wie vor am Rennen teil und kann Ihnen den Weg abschneiden! Dieser Modus ist nur etwas für Kamikaze-Fahrer...

<F4>: Damit bestimmen Sie die Geschwindigkeit. Voreingestellt ist »9« (High Speed). Bis zur niedrigsten Stufe »1« (sehr lahm) läßt sie sich zurückstellen.

<F5>: macht das Gitter in den Spielfeldern unsichtbar.

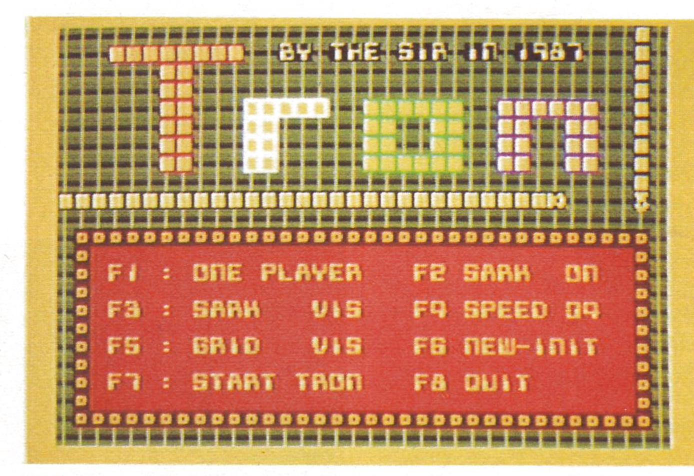
<F6>: Alle bisherigen Einstellungen werden rückgängig gemacht. Das Spielmenü erscheint wie nach RUN.

<F7>: startet das Spiel.

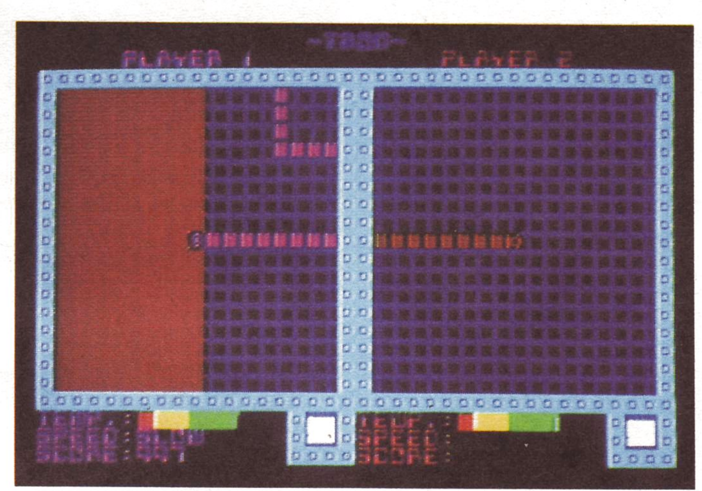
<F8>: Man kann »Tron« mit einem Reset verlassen.

Wenn Sie allein gegen den Computer spielen möchten, müssen Sie den Joystick in Port 1 stecken. Das linke Spielfeld (Abb. 2) gehört Ihnen, auf dem rechten treibt Sark sein Unwesen (vom Computer gesteuert). Spielen Sie mit einem Freund, muß der seinen Joystick im Port 2 anstöpseln.

Damit's nicht zu leicht wird, sehen beide Spieler nur jeweils einen begrenzten Ausschnitt des Spielfelds. Vorsicht: Plötzlich kann vor Ihnen der Spielfeldrand oder die gegnerische Energiebarriere auftauchen! Dann heißt's schnell reagieren und per Joystick einen Haken schlagen, sonst – Fluß ohne Wiederkehr. Außerdem sollten Sie vermeiden, sich den Ausgang mit eigenen Barrieren zu verbauen.



[1] Im Startbildschirm von »Tron« lassen sich die Spielmodi einstellen



[2] Beim »One-Player«-Modus gehört Ihnen das linke Spielfeld

Kurzinfo: Tron

Programmart: Strategiespiel
Laden: LOAD "TRON",8
Starten: nach dem Laden RUN eingeben
Steuerung: Joystick Port 1 (Spieler 1) und 2 (Spieler 2)
Besonderheiten: wahlweise für einen oder zwei Spieler, neun verschiedene Geschwindigkeitsstufen
Benötigte Blocks: 27
Programmautor: Christian Dombacher

Strategisch denken, schnell reagieren!

Damit Sie den Überblick behalten, hilft Ihnen die »Radar-Sonar-Anzeige« am rechten unteren Rand beider Spielfelder: Ein winziger Punkt zeigt die ungefähre Position der Laser-Bykes. Wer Gas geben will, muß auf den Feuerknopf drücken. Die Anzeige »Speed« gibt Auskunft über den aktuellen Geschwindigkeitsmodus. Gesammelte Punkte aus der letzten Runde werden übernommen und nicht gelöscht. Eine High score-Liste ist im Spiel jedoch nicht vorgesehen.

Wer Gas geben will, sollte den Feuerknopf drücken. Dazu muß man aber die Temperaturanzeige des Kommandopults im Auge behalten, sonst gibt's unweigerlich einen Motorschaden (wegen Überhitzung!). Sie steigt vom grünen in den gelben und anschließend in den roten Bereich: Dann ist's aber meistens schon zu spät!

Ziel jedes Spielers ist, so lange wie möglich im Rennen zu bleiben und dem Gegner bis an die Schmerzgrenze einzuheizen.

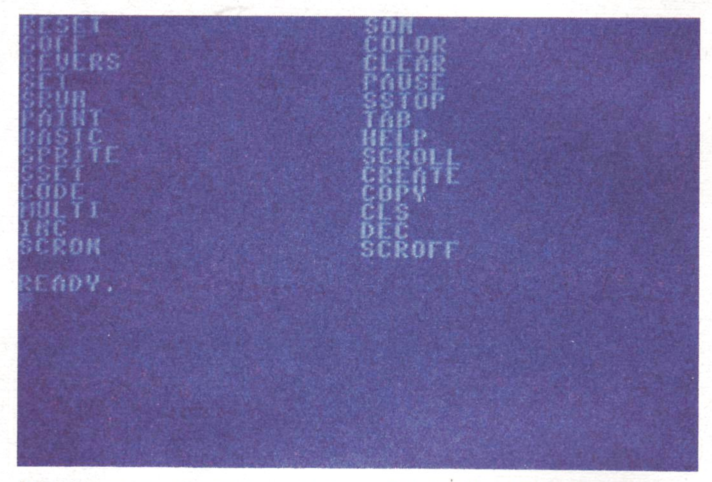
Trost: Wenn's doch einmal schief gegangen ist, drücken Sie <F7> – und Sie können mit einem neuen Flitzer weitermachen!

(Christian Dombacher/bl)

fast aus dem Handgelenk

Game-Basic – Spiele selbst gemacht

Tolle Spiele mit geringem Programmieraufwand? Kein Problem für »Game-Basic«! Effektvoll lassen sich Sprites und Bildschirm-Scrolling realisieren.



[1] Alle neuen Befehle auf einen Blick: die HELP-Funktion bringt die neuen Anweisungen auf den Bildschirm

Eines ist klar: Super-Games wird man ausschließlich unter Verwendung von Maschinensprache zustande bringen (s. unser großer Spielkurs in diesem Heft).

Und dennoch: Wem Assembler ein Buch mit sieben Siegeln ist, muß deswegen nicht seinen Programmierdrang unterdrücken: Unsere Basic-Erweiterung »Game-Basic« stellt 26 neue Basic-Befehle zur Verfügung (Tabelle). Mit etwas Geschick und Basic-Programmierwissen können Sie Ihrer Fantasie freien Lauf lassen und (fast) jede Spielidee verwirklichen.

Laden Sie das Programm mit:

```
LOAD "GAME-BASIC",8
```

Gestartet wird mit RUN.

Auf dem Bildschirm erscheint die neue Einschaltmeldung: Der C 64 steht für Ihre Programmeingaben bereit. Alle wichtigen Adressen für Bildschirm, Farb-RAM, Sprites, Soundchip, Timer usw. bleiben unverändert. Der Bereich $0800 (2048) bis $0FFF (4095) ist für 32 Sprite-Muster reserviert. »Game-Basic« selbst braucht den Speicher ab Adresse $1000 (4096) bis $17B0 (6064). Der dahinterliegende Platz ab $17B1 (6065) bis $1FFF (8191) bleibt frei und läßt sich für eigene Maschinenprogramme nutzen. Der Zeichensatz wurde ab $2000 (8192) plaziert. Dort läßt er sich problemlos ändern (Zeichen umdefinieren). Der Basic-Speicher für Ihre Spielprogramme beginnt jetzt bei $2800 (10240) und erstreckt sich – wie gewohnt – bis $9FFF (40959). Die RAMs unterm Interpreter- und Kernel-ROM werden von »Game-Basic« nicht benutzt. Achtung: Das Programm arbeitet ausschließlich im Textmodus: Es wird keine hochauflösende Grafik verwendet! Normalerweise sind Spiele nicht mit Hires-Grafik ausgestattet, sondern mit verändertem Zeichensatz (Spiellandschaften und Levels) – das gilt auch für die meisten professionellen Spiele.

Die neuen Befehle

Alle Programme, die mit Befehlen des Basic 2.0 entwickelt wurden, laufen anstandslos mit der Basic-Erweiterung.

»Game-Basic« stellt für die neuen Anweisungen einprägsame Namen zur Verfügung, die sich nahezu selbst erklären. Sie können im Direkt- und Programm-Modus angewandt werden:

HELP

Gerade bei Basic-Erweiterungen ist es hilfreich, die Schreibweise der neuen Befehle ständig zur Hand zu haben (bis man sich daran gewöhnt hat!). Alle neuen Befehle erscheinen auf dem Bildschirm (Abb. 1).

BASIC

Diese Anweisung deaktiviert »Game-Basic«. Der C 64 verhält sich wieder normal. Mit »SYS 4096« läßt sich die Erweiterung erneut aktivieren. Achtung: Das gilt nur, solange dieser Speicherbereich nicht von anderen Programmen oder Daten überschrieben wurde!

RESET

Die Basic-Erweiterung wird neu gestartet: Es erscheint wieder die Einschaltmeldung. Alle veränderten Einstellungen (Farben, Sprites usw.) werden in den Originalzustand zurückversetzt. Ein Basic-Programm im Speicher bleibt dabei ungelöscht.

COLOR

ändert die Rahmen-, Hintergrund- und Zeichenfarbe gemäß den entsprechenden Parametern, die hinter COLOR stehen müssen (s. Tabelle). Es gelten die Farbcodes von 0 bis 15. Die Anweisung ersetzt folgende Befehle:

```
POKE 53281,Farbe
POKE 53280,Farbe
POKE 646,Farbe
```

Ein Beispiel für blauen Rahmen, weißen Hintergrund und grüne Schriftzeichen:

```
COLOR 6,1,5
```

SCROFF

schaltet den Bildschirm vorübergehend ab (während z.B. in der Zwischenzeit eine Bildschirm-Grafik durch Ihr Programm aufgebaut wird).

SCRON

Der Bildschirm wird wieder eingeschaltet.

Die beiden letztgenannten Befehle ersparen umständliche POKE-Anweisungen zur Manipulation des VIC-Chips.

PAINT

füllt den Farbspeicher fürs Bildschirm-RAM mit der angegebenen Farbe. Selbstverständlich kann man die Zeichen auf dem Bildschirm auch jederzeit nachträglich umfärben. Damit lassen sich reizvolle Effekte erzielen.

TAB

schaltet die übersichtliche Ausgabe eines Listings auf dem Bildschirm an oder aus. Bei Programmstart ist die Funktion aktiviert (Abb. 2). Jeder Befehl wird in eine separate Bildschirmzeile geschrieben.

Achtung: Bei eingeschalteter TAB-Funktion läßt sich kein Basic-Listing vernünftig editieren – dazu muß sie abgeschaltet werden (dies erledigt ebenfalls der Befehl TAB!).

INC und DEC

»Inkrement« (Erhöhen) und »Dekrement« (Verringern) fungieren als Abwandlung des POKE-Befehls: Im Gegensatz zum POKEn wird kein bestimmter Wert in eine Speicherstelle geschrieben, sondern der darin befindliche Inhalt um den Parameter »eh« erhöht bzw. erniedrigt.

Beispiel: Der Inhalt von Speicherstelle 49152 soll um »12« erhöht werden:

```
INC 49152,12
```

PAUSE

ersetzt die umständliche Zeitschleife des Basic 2.0:

```
FOR T=1 TO 1000: NEXT T
```

Damit wird der Programmablauf ca. 1 s angehalten. Bei »Game-Basic« lautet die Anweisung:

```
PAUSE 500
```

Bildschirmzeichen ändern

Jedes Spiel mit Profi-Touch kommt nicht ohne geänderten Zeichensatz aus. »Game-Basic« stellt auch dazu komfortable Anweisungen zur Verfügung:

CREATE

teilt dem C64 mit, welches Zeichen Sie ändern möchten. Dazu ist es nötig, den Modus »1« anzugeben. Beispiel:

Sie wollen aus dem »O« den Umlaut »Ö« machen.

```
CREATE 1,15
```

Der Modus »1« gilt für Zeichenänderungen, »0« bedeutet Sprite-Definitionen (s. dort). »15« ist der Bildschirmzeichen-Code für den Buchstaben »O«.

CODE

leitet die neue Bitmuster-Vergabe fürs neue Zeichen ein (unser Beispiel: Ö). Ein Punkt gilt als gelöschtes Pixel, die »1« signalisiert, daß

[2] Übersichtliche Ausgabe eines Basic-Listings mit TAB. Diese Basic-Zeilen lassen sich nicht editieren.

Kurzinfo: Game-Basic

Programmart: Basic-Erweiterung
Laden: LOAD "GAME-BASIC",8
Starten: nach dem Laden RUN eingeben
Besonderheiten: Der Befehl COPY überträgt den Zeichensatz vom ROM ins RAM. Sprite-Bewegung im Interrupt.
Benötigte Blocks: 9
Programmautor: Ronald Meyer

COPY

verlegt den Originalzeichensatz vom ROM ins RAM (ab Adresse $2000). Es ist auch möglich, nach dem COPY-Befehl einen geänderten Zeichensatz (nur Großschrift/Blockgrafik, 2048 Bytes) nachzuladen. Achtung: Die Startadresse dieser Zeichensatzdateien auf Diskette muß $2000 (8192) sein!

Die Änderung eines beliebigen Zeichens läßt sich im Basic-Programm realisieren:

der entsprechende Bildpunkt eingeschaltet ist.

```
10 CREATE 1,15
20 CODE .11..11.
30 CODE ........
40 CODE ..1111..
50 CODE .11..11.
60 CODE .11..11.
70 CODE .11..11.
80 CODE .11..11.
90 CODE ..1111..
```

Da viele Bitmuster-Zeilen mit dem Original-»O« übereinstimmen, würden auch folgende Änderungen genügen:

```
10 CREATE 1,15
20 CODE .11..11.
30 CODE ........
40 CODE ..1111..
```

Ein weiteres Beispiel: Der Klammeraffe soll künftig als Copyright-Zeichen erscheinen.

```
10 CREATE 1,0
20 CODE .111111.
30 CODE 1......1
40 CODE 1..11..1
50 CODE 1.1....1
60 CODE 1.1....1
70 CODE 1..11..1
80 CODE 1......1
90 CODE .111111.
```

Nach der Tastenkombination <RUN/STOP RESTORE> oder der Anweisung RESET erscheinen allerdings die Zeichen wieder normal: Also Hände weg von diesen Tasten!

Selbstverständlich läßt sich der Zeichensatz im RAM auch durch die altbewährte POKE-Methode (mit DATA-Werten) verändern. Beachten muß man dabei allerdings, daß für einen neuen Zeichensatz nur die Großschrift/Blockgrafik-Zeichen zur Verfügung stehen (neun Blöcke auf Diskette!)

MULTI

Dieser Befehl gilt ebenfalls für Zeichen oder Sprites. Das regelt der Modus-Parameter »c« (s. Tabelle). Jetzt können noch zwei Zusatzfarben bestimmt werden. Ein Beispiel:

```
MULTI 1,5,2,6,7
```

Ab sofort erscheinen alle Zeichen auf dem Bildschirm in Grün (5) und Rot (2). Sprites dagegen besitzen die Farben Blau (6) und Gelb (7). Dennoch ist unumgänglich, das entsprechende Zeichen mit CREATE und CODE vorzubereiten. Eine farbige Version des kleinen »ö« sieht z.B. so aus (wir haben das englische Pfundzeichen dazu verwendet):

```
5 MULTI 1,5,2
10 CREATE 1,28
20 CODE ..11..11
30 CODE ..11..11
40 CODE .......
50 CODE ..222222
60 CODE ..22..22
70 CODE ..22..22
80 CODE ..33..33
90 CODE ..333333
```

Die Codes der Bitmuster:
- 1: Zusatzfarbe 1,
- 2: Zusatzfarbe 2,
- 3: aktuelle Zeichenfarbe.

Achten Sie darauf, daß die Codezahlen der Bitmuster immer paarweise angegeben werden (für Multicolor unabdingbar!). Sonst erhalten Sie eine Fehlermeldung.

Sprites entwerfen

Ähnlich wie bei den Zeichen spielt sich bei »Game-Basic« auch die Entwicklung von Sprites ab. Eine durch die Normalkonfiguration des VIC-Chips bedingte Einschränkung gilt hier ebenfalls: Es lassen sich nicht mehr als acht Sprites gleichzeitig auf dem Bildschirm aktivieren!.

Bei CREATE ist nun der Modus »0« und der entsprechende Speicherbereich (Sprite-Block) anzugeben, in dem das Sprite-Muster abgelegt werden soll. Sie können die Blöcke 32 bis 63 verwenden. Den Beginn des Sprite-Blocks im Speicher erhalten Sie, wenn Sie die Blocknummer mit »64« multiplizieren. Außerdem besitzt eine Bit-Zeile für die CODE-Anweisung jetzt 24 Stellen (statt acht). 21 einzelne Bit-Zeilen müssen entworfen werden. Ein Beispiel: Sie wollen ein Sprite ab Adresse 2048 (Spriteblock 32) definieren:

```
10 CREATE 0,32
20 CODE ........ usw.
```

Im Listing unseres Programmbeispiels »Game-Basic-Game« finden Sie eine Menge Sprite-Definitionen, deren Aufbau man als Hinweis für eigene Kreationen betrachten sollte.

SPRITE

definiert ein Sprite, dessen Muster im entsprechenden Block steht. Folgende Parameter sind wichtig:
- Sprite-Nummer,
- Block,
- gedehnt in y-Richtung (vertikal),
- in x-Richtung vergrößert,
- Priorität vor oder hinter den Zeichen auf dem Bildschirm,
- Multicolor,
- Grundfarbe.

SSET

positioniert ein Sprite an gewünschter Stelle auf dem Bildschirm. Horizontale Werte von 0 bis 511 und ver-

tikale von 0 bis 255 sind möglich. »Game-Basic« berücksichtigt dabei automatisch das höchstwertige Bit im VIC-Register 16, das ab Bildschirmposition 256 Bedeutung erlangt. Beachten Sie, daß man Sprites erst dann auf dem Bildschirm sieht, wenn sie mindestens 24 Pixel von diesen Grenzen entfernt sind! Dazu geben Sie die Sprite-Nummer sowie die x- und y-Position an. Beispiel: Etwa in der Bildschirm-Mitte soll Sprite Nr.5 erscheinen.

```
SSET 5,184,142
```

SON

Das Sprite mit einer anzugebenden Nummer wird eingeschaltet.

SOFF

verbannt ein Sprite mit entsprechender Nummer wieder vom Bildschirm.

SRUN

Damit bewegt sich ein Sprite (oder mehrere) in eine von acht möglichen Richtungen. Der Interrupt übernimmt ab sofort die Sprite-Steuerung: Ihr Programm wird dadurch nicht aufgehalten. Damit kann ein Sprite während der Bewegung die Gestalt verändern (z.B. explodieren). Sie können auch das Programm mit der STOP-Taste abbrechen: Die Sprites setzen ihre Bahn unbeirrt fort (da hilft nur <RUN/STOP RESTORE) oder

SSTOP

Auch hier muß die Sprite-Nummer angegeben werden. Dieser Befehl befiehlt dem Kobold auf dem Bildschirm, sofort anzuhalten. Möchten Sie alle acht Sprites stoppen, benutzen Sie:

```
SSTOP 8
```

Komfortable Bildschirmverwaltung

»Game-Basic« verurteilt die inversen Cursor-Steuerzeichen des Commdore-Zeichensatzes zur Untätigkeit:

SET

ist ein erweiterter PRINT-Befehl zur Textausgabe an gewünschter Stelle auf dem Bildschirm. Möchten Sie z.B. das Wort »Text« mit weißen Buchstaben in der 18. Spalte und der 12. Zeile plazieren, lautet die Anweisung:

```
SET 18,12,1,"TEXT"
```

Mit dem üblichen PRINT-Befehl wären eine Unmenge Cursor-Steuerzeichen nötig. Außerdem ist die SET-Anweisung bedeutend schneller.

REVERS

Der aktuelle Inhalt des Bildschirms wird invers gezeigt.

CLS

löscht den Bildschirm. Dieser Befehl ersetzt die Anweisung:

```
PRINT CHR$(147)
```

CLEAR

entfernt einzelne Zeilen oder Blöcke, entsprechend Ihrer Parameterangaben, vom Screen.

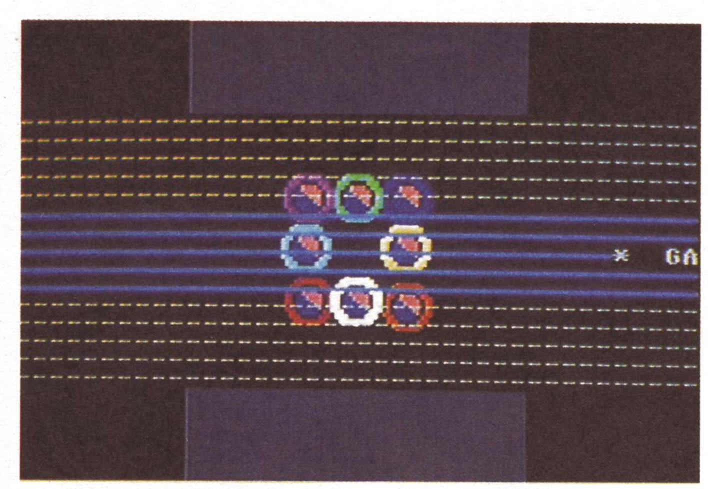
Ein Beispiel: Der Bildschirm soll zwischen der achten und einschließlich 13. Zeile gelöscht werden.



[3] Beispielprogramm »Gamebasic-Game«: Retten Sie die Fallschirmspringer!

Game-Basic (Befehlsübersicht)

Befehl	Bedeutung
RESET	Neustart der Basic-Erweiterung. Programme im Speicher werden dabei nicht gelöscht.
HELP	bringt alle neuen Befehle auf den Bildschirm
BASIC	Game-Basic wird ausgeschaltet. Der Normalmodus ist wieder aktiv (Basic 2.0).
COLOR rf,hf,zf	Rahmen- (rf), Hintergrund- (hf) und Zeichenfarbe (zf) setzen
REVERS	invertiert den aktuellen Bildschirm
PAINT f	färbt den Bildschirminhalt mit der Farbe f (Werte von 0 bis 15)
CLS	Bildschirm löschen
SCROFF	Bildschirm abschalten
SCRON	schaltet Bildschirm wieder ein
CLEAR an,en	Bildschirm wird ab Zeile »an« bis Zeile »en« gelöscht
COPY	Zeichensatz ins RAM kopieren
TAB	schaltet formatierten LIST-Modus ein/aus
PAUSE t	Das Programm hält t x 0,002 s an. Beispiel: »PAUSE 500« unterbricht den Programmablauf eine Sekunde.
INC ad,eh	addiert den Wert »eh« zum Inhalt der Adresse »ad«
DEC ad,eh	subtrahiert die Zahl »eh« vom Wert in der Speicherstelle »ad«
SET s,z,f,"text"	gibt eine Zeichenkette »text« in der Spalte »s« und der Zeile »z« in der Farbe »f« aus.
SCROLL az,ez,s,z,f,r	scrollt den Bildschirmbereich von der Anfangszeile »az« bis zur Endzeile »ez« in Richtung »r« (0 = rechts, 1 = links). Dabei werden Zeichen mit dem Code »z« in der Farbe »f« nachgescrollt.
SON s	schaltet Sprite mit der Nummer »s« ein (0 bis 7)
SOFF s	das Sprite mit der Nummer »s« wird wieder ausgeschaltet
SPRITE s,bk,ye,xe,pr,mc-,f	bestimmt das Aussehen von Sprite Nummer »s«: - bl = Block (32 bis 64), - ye = doppelte Höhe (0 = nein, 1 = ja), - xe = doppelte Breite (0 = nein, 1 = ja), - pr = Priorität vor (= 0) oder hinter (= 1) dem Bildschirminhalt, - mc = Multicolor an (= 1) oder aus (= 0), - f = Spritefarbe.
SSET s,x,y	setzt Sprite Nummer »s« an Position x (0 bis 511!) und y (0 bis 255)
SRUN s,ri	Sprite »s« bewegt sich in die angegebene Richtung:
SSTOP s	Sprite »s« hält wieder an
MULTI c,z1,z2,s1,s2	Multicolormodus ein-/ausschalten: - c: 0 = Mulitcolor für Zeichen an, 1 = Multicolor für Zeichen aus, 2 = Angabe von Zusatzfarben, Bei Verwendung von 0 und 1 muß man die Zusatzfarben nicht angeben. - z1: Zusatzfarbe 1 (Zeichen), - z2: Zusatzfarbe 2 (Zeichen), - s1: Zusatzfarbe 1 (Sprites), - s2: Zusatzfarbe 2 (Sprites).
CREATE m,bf	neues Zeichen oder Sprite definieren: - m: Mode (0 = Sprite, 1 = Zeichen), - b1: Spriteblock oder Zeichencode.
CODE Bitfolge	Nur in Zusammenhang mit dem CREATE-Befehl anwenden! Bei Sprites muß man 24, bei Zeichen acht Bits eintragen. - . = Hintergrundfarbe, - 1 = Farbe des Zeichens/Sprites (Multicolormodus Zusatzfarbe 1), - 2 = Zusatzfarbe 2, - 3 = Farbe des Zeichens/Sprite bei Multicolormodus.



[4] Unser Demoprogramm zeigt die komfortablen Funktionen der neuen Basic-Befehle

```
CLEAR 7,12
```

Beachten Sie, daß die Zeilenwerte zwischen »0« (1. Zeile) und maximal »24« (25. Zeile) liegen dürfen.

SCROLL

verschiebt oder verändert beliebige Bildschirmbereiche. Eine nähere Beschreibung der Parameter finden Sie in der Tabelle. Mit dem Befehl können z.B. Laufschriften simuliert werden.

Demoprogramme auf Diskette

Auf der Diskette zu diesem Sonderheft gibt's ein Basic-Spiel, das ausschließlich mit den neuen Befehlen dieser Befehlserweiterung konzipiert wurde: »Gamebasic-Game«. Achten Sie darauf, daß »Game-Basic« aktiv ist, denn das normale Basic 2.0 versteht die neuen Befehle nicht.

Laden Sie das Programm mit:

```
LOAD "GAMEBASIC-GAME",8
```

und starten Sie es mit RUN. Ein neuer Zeichensatz (ZSA) wird nachgeladen.

Die Spielidee ist rasch erklärt: Sie sind Kapitän eines Flugzeugträgers, der auf hoher See aus Düsenjägern abspringende Fallschirmjäger aufnehmen soll (Abb. 3). Das Schiff befindet sich am unteren Bildschirmrand in stürmischer See und läßt sich mit dem Joystick in Port 2 nach links und rechts steuern. Achtung: Die Reaktion des Joysticks wurde absichtlich erschwert, damit das Spiel nicht zu einfach ist. Oben zischen in rascher Folge die Düsenjäger vorbei und werfen die Fallschirmspringer ab, die Sie durch geschicktes Manövrieren des Schiffs wohlbehalten aufnehmen müssen.

Am oberen Bildschirmrand sieht man unter der Anzeige »Score«, wie viele Springer sicher gelandet sind. Wenn einer den Kreuzer verfehlt, verlieren Sie ein Leben (Anzeige »Lives«). Bei Programmstart stehen Ihnen drei zur Verfügung.

Wenn die Spielrunde zu Ende ist, scrollt der Bildschirm nach rechts. Mit der Taste <J> können Sie ein neues Spiel beginnen, bei <N> führt »Game-Basic« den Befehl RESET aus: Das Spiel steht allerdings unversehrt im Speicher und kann jederzeit erneut mit RUN gestartet werden.

In den Basic-Zeilen 100 bis 810 werden mit den neuen Befehlen CREATE und CODE vier Sprite-Muster definiert:

- Zeile 100 bis 250: Flugzeug von links,
- Zeile 260 bis 410: Flugzeug von rechts,
- Zeile 420 bis 630: Fallschirmspringer,
- Zeile 640 bis 810: Schiff.

Die Zeichen CHR$(164) und CHR$(165) ändern durch die Basic-Zeilen 820 bis 1000 ihr Aussehen: Sie dienen jetzt als Füllmuster für die Wellenbewegung des Meeres, die mit der Anweisung SCROLL erzeugt wird (Zeile 1170). Den stetigen Flug der Düsenjäger über den Bildschirm bewirkt der Befehl »SRUN« in den entsprechenden Basic-Zeilen. Im Bereich 1050 bis 1100 werden die Sprites eingeschaltet (SPRITE), ab Zeile 1110 positioniert (SSET). Daß die Sprites bei Erreichen der bekannten Position »256« nicht vom Bildschirm verschwinden, verdanken wir den Befehlen in den Programmzeilen 1190 bis 1200. Dabei bildet die Variable »V« die Basic-Adresse des VIC-Chip: 53248. Die Joystick-Abfrage zur Bewegung des Rettungskreuzers nach links oder rechts beginnt ab Zeile 1210. Der gesamte Spielablauf (Sprite-Bewegung, Wellengang, Regulierung der Anzeigen »Score« und »Lives«) wird im Prinzip nur durch die Zeilen 1170 bis 1280 geregelt: Darin liegt der Kern des Spielprogramms. Wenn Sie bei Programmende die Frage nach einem neuen Spiel verneinen, schiebt die SCROLL-Anweisung in Zeile 1330 den gesamten Bildschirm nach rechts und kehrt mit RESET zum Einschaltbildschirm von »Game-Basic« zurück.

Falls Ihnen die eingestellten Werte für »Score« und »Lives« zu gering sind, läßt sich das jederzeit ändern: Die Anzahl der verfügbaren Leben steht in der Variablen »L« (Programmzeile 1160). Soll der Score bei jeder erfolgreichen Bergung eines Fallschirmspringers um z.B. 1000 Zähler klettern, müssen Sie die Addition in Zeile 1270 um »1000« erhöhen (statt nur um »1«):

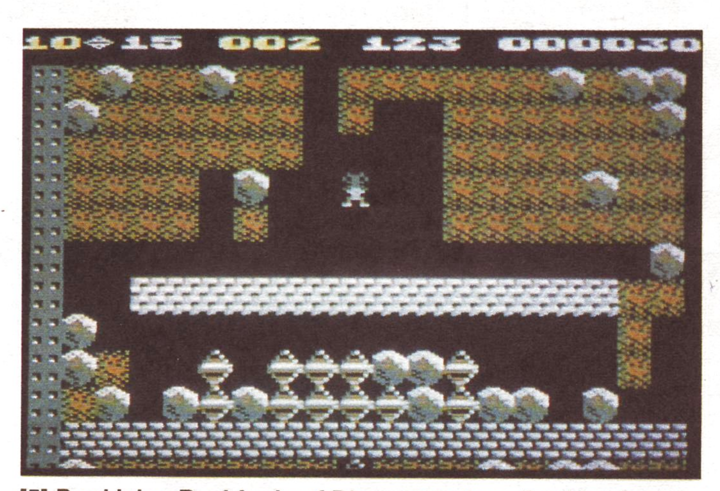
```
IF Y=>48 THEN SC=SC+1000
```

Eine Fülle von Bildschirm- und Sprite-Manipulationen zeigt Ihnen das nächste Beispielprogramm:

```
LOAD "GAMEBASIC-DEMO",8
```

Es bietet nahezu alle Funktionen, die »Game-Basic« zur Verfügung stellt: Sprite-Bewegung in alle Richtungen, Bildschirm-Scrollen, Farbwechsel und raffiniertes Löschen bestimmter Bildschirmbereiche (Abb. 4).

Studieren Sie aufmerksam die Listings der Demoprogramme und experimentieren Sie mit »Gamebasic-Game« (denkbar wäre statt eines Schiffs auch ein Lastwagen, statt Flugzeugen könnten z.B. große Vögel Früchte abwerfen, die der Trucker aufsammeln muß). Dazu müssen nur die Sprite-Definitionen in den entsprechenden Zeilen geändert und das Programm neu gespeichert werden.



[5] Der kleine Rockford auf Diamantsuche: »Boulder Dash« ist einer der vielen Spiele-Klassiker des C 64!

Ihrer Fantasie sind keine Grenzen gesetzt: Mit »Game-Basic« werden selbstprogrammierte Spiele zum Vergnügen! Diese Basic-Erweiterung bietet viele Möglichkeiten, auch professionell funktionierende Spiele zu entwickeln. Wie wär's z.B. mit einer eigenen Version von »Boulder Dash« (Abb. 5)? Viel Vergnügen beim Programmieren!

(Ronald Meyer/bl)

SONDER
HEFT
74
VORSCHAU

■ - Endlich steht er auf dem Tisch, der heißersehnte C64! Unser Einsteiger-Sonderheft Nr. 74 demonstriert auch völlig Unbeleckten verständlich, welche ungeahnten Möglichkeiten in diesem Computer stecken:

■ - »Basic V3.5« mit mehr als 40 neuen Befehlen, bietet alles für Softwarespezialisten in spe: Programmierhilfen, hochauflösende Grafik und komfortable Soundanweisungen.

■ - Es muß nicht immer Geos sein: Mit »F.O.B.S«, dem fensterorientierten Betriebssystem, werden Diskettenoperationen zum Kinderspiel.

■ - Außerdem finden Sie jede Menge Hilfen, Tips & Tricks, PEEKs & POKEs: den Goldenen Schlüssel zur Welt des C64!

■ - Eine ausführliche Marktübersicht bringt viele Tips zu Software und weiterführender Literatur.

EINSTEIGER
64'er
DISKETTE IM HEFT

Das Sonderheft 74 gibt's ab 24.1.92 bei Ihrem Zeitschriftenhändler.

Aus aktuellen oder technischen Gründen können Themen verschoben werden. Wir bitten dafür um Verständnis.

Mitmachseiten

Leserideen sind uns wichtig

Das Echo auf unsere Mitmach-Aktion im 64'er-Sonderheft 67 war überwältigend! Heute stellen wir die Gewinner und die meistgenannten Wünsche unserer Leser vor.

Als sich unsere Redakteure durch den Stapel der eingesandten Mitmachseiten gearbeitet hatten, waren sie zwar »jenseits von gut und böse«, aber doch um etliches schlauer. Nützliche Vorschläge, interessante Beurteilungen und Bewertungen: alles wurde gesammelt.

In den künftigen Ausgaben der 64'er-Sonderhefte werden wir uns bemühen, Ihre Ideen zu verwirklichen.

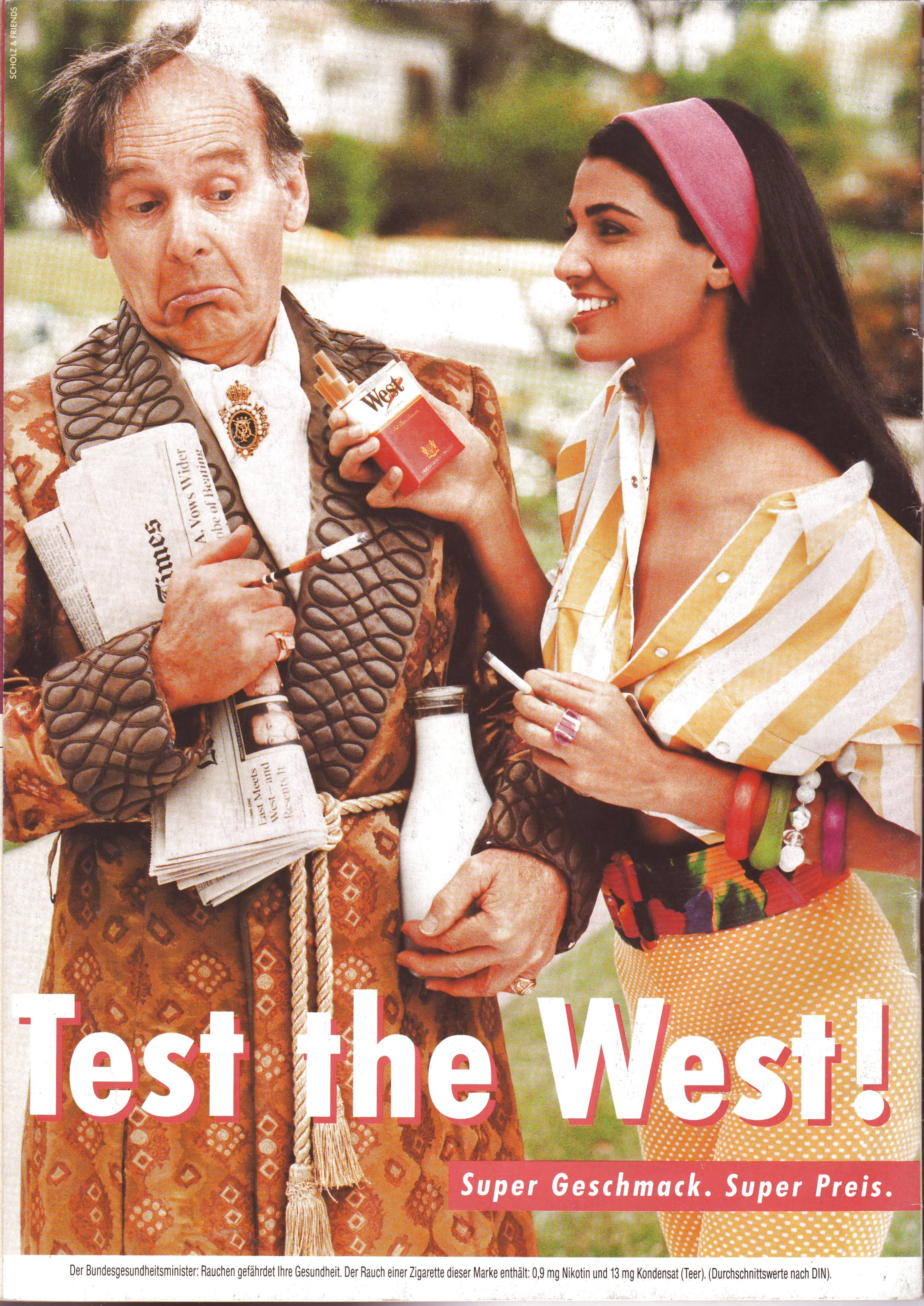
Das ist die Leserwunsch-Hitparade zur Themengestaltung der Sonderhefte (prozentualer Anteil an der Gesamtzahl der Einsendungen dahinter):

Unsere Themen im Spiegel des Interesses

groß	durchschnittlich	niedrig	
15%	40%	45%	Spiele
50%	43%	7%	Grafik
70%	25%	5%	Anwendungen
77%	18%	5%	Tips & Tricks
71%	26%	3%	Hardware
27%	50%	23%	Basic
47%	31%	22%	Assembler
51%	37%	12%	Programmieren
55%	38%	7%	Floppy
62%	28%	10%	Drucker
40%	35%	25%	Sound
25%	30%	45%	Einsteiger

Als Belohnung für Ihre Mühe verlosten wir zehn 64'er-Sonderhefte unter den Einsendern. Unsere Glücksfee Sylvia hat in den Stapeln gewühlt und zehn Gewinner herausgefischt:

Harry Schröder, 4047 Dormagen, SH36
Johann Haberl, A4863 Seewalchen, SH53
Peter Wegener, 5090 Leverkusen 1, SH53
Wilfried Jellasics, 3600 Genk/Belgien, SH35
Michael Zimmer, 6625 Püttlingen, SH56
Günter Franke, 3340 Wolfenbüttel, SH 22
Karl-Ludwig Pauls, 5303 Bornheim 3, SH57
Steffen Warzecha, 7050 Leipzig, SH68
Klaus Collatz, 3241 Flechtingen, SH48
Udo Weidmann, 4500 Solothurn/Schweiz, SH31



SCHOLZ & FRIENDS
Times
A. Vows Wider
East Meets
West – and
Resents It
West
Test the West!
Super Geschmack. Super Preis.
Der Bundesgesundheitsminister: Rauchen gefährdet Ihre Gesundheit. Der Rauch einer Zigarette dieser Marke enthält: 0,9 mg Nikotin und 13 mg Kondensat (Teer). (Durchschnittswerte nach DIN).